

VYSOKÁ ŠKOLA BÁŇSKÁ – TECHNICKÁ UNIVERZITA OSTRAVA
EKONOMICKÁ FAKULTA

KATEDRA SYSTÉMOVÉHO INŽENÝRSTVÍ

Návrh rezervačního systému pro podnikatele z oblasti sportu a relaxace

The Design of Booking System for the Entrepreneurs in the Field of Sport and
Relaxation

Student: Bc. Pavlína Polanská

Vedoucí diplomové práce: Ing. Radek Němec

Ostrava 2013

Zadání diplomové práce

Student: **Bc. Pavlína Polanská**

Studijní program: N6209 Systémové inženýrství a informatika

Studijní obor: 6209T025 Systémové inženýrství a informatika

Téma: **Návrh rezervačního systému pro podnikatele z oblasti sportu a relaxace**
The Design of a Booking System for Entrepreneurs in the Field of Sport and Relaxation

Zásady pro vypracování:

1. Úvod
2. Teoretická a metodická východiska objektově orientované analýzy návrhu informačního systému
3. Analýza současného stavu
4. Návrh rezervačního systému pro podnikatele
5. Závěr

Seznam použité literatury

Seznam zkratk

Prohlášení o využití výsledků diplomové práce

Seznam příloh

Přílohy

Seznam doporučené odborné literatury:

GEORGE, Joey F., Dinesh BATRA, Joseph S. VALACICH a Jeffrey A. HOFFER. *Object-oriented Systems Analysis and Design*. 2nd ed. New Jersey: Pearson, 2007. 464 s. ISBN 978-0132279000.

GRÄSSLE, Patrick, Henriette BAUMANN a Philippe BAUMANN. *UML 2.0 in action - a project based tutorial*. Birmingham: Packt Publishing, 2005. 229 s. ISBN 978-1-904811-55-8.

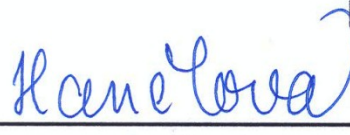
MERUNKA, Vojtěch. *Objektové modelování*. Praha: Alfa Nakladatelství, 2008. 184 s. ISBN 978-80-87197-04-2.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

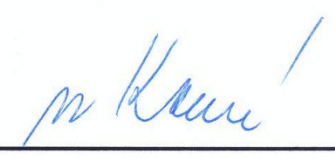
Vedoucí diplomové práce: **Ing. Radek Němec**

Datum zadání: 23.11.2012

Datum odevzdání: 26.04.2013


doc. Ing. Jana Hančlová, CSc.
vedoucí katedry




prof. Dr. Ing. Dana Dluhošová
děkanka fakulty

„Chtěla bych mimořádně poděkovat svému vedoucímu diplomové práce Ing. Radku Němcovi za ochotu, vstřícnost a cenné rady během tvorby mé diplomové práce. Dále bych chtěla poděkovat Petru Bažantovi za poskytnutí externí konzultace a užitečného náhledu do praxe. Poděkování patří rovněž mé rodině, která mě ochotně podporovala po celou dobu tvorby mé diplomové práce.“

„Místopřísežně prohlašuji, že jsem celou diplomovou práci včetně všech příloh vypracovala samostatně.“

V Ostravě 26. 04. 2012



.....
Bc. Pavlína Polanská

Obsah

Obsah	3
1. Úvod.....	5
2. Teoretická a metodická východiska objektově orientované analýzy a návrhu informačního systému	6
2.1 Softwarové inženýrství	7
2.1.1 Vodopádový přístup	7
2.1.2 Iterativní přístupy	9
2.2 Objektově orientované metodiky a notace.....	9
2.2.1 Některé starší metodiky a notace objektově-orientovaného modelování ..	9
2.2.2 Jazyk UML a Unified process	13
2.3 Analýza a návrh systému pomocí UML	26
2.3.1 Specifikace požadavků (Inženýrství požadavků)	26
2.3.2 Tvorba případů užití	29
2.3.3 Tvorba tříd	30
2.3.4 Diagramy aktivit	31
2.3.5 Realizace případů užití – Diagramy interakcí	31
2.3.6 Diagramy stavového automatu	32
2.4 Shrnutí.....	33
3. Analýza současného stavu.....	34
3.1 Analýza služeb sportcentra	36
3.1.1 Služby jednorázové	36
3.1.2 Služby celodenní	36
3.2 Analýza cen a permanentek sportcentra.....	37
3.2.1 Vstupové permanentky	37
3.2.2 Kreditové permanentky	37

3.2.3	Časové permanentky	38
3.3	Shrnutí a strukturování funkčních požadavků na rezervační systém	38
4.	Návrh rezervačního systému pro podnikatele	41
4.1	Modelování případů užití	41
4.1.1	Aktéři systému	41
4.1.2	Případy užití	41
4.2	Návrh tříd	70
4.3	Diagramy aktivit	77
4.4	Sekvenční diagramy	80
4.5	Diagramy stavového automatu	82
4.6	Návrh uživatelského rozhraní rezervačního systému	84
5.	Závěr	85
	Seznam použité literatury	86
	Prohlášení o využití výsledků diplomové práce	88
	Seznam příloh	89

1. Úvod

Využití rezervačních systémů v podnikání je v dnešní době již poměrně běžná záležitost. Tyto systémy usnadňují a zefektivňují práci zaměstnancům a zároveň jsou velice přívětivou službou pro zákazníky. Jednou z oblastí, kde lze tyto systémy poměrně hojně využívat jsou bezpochyby relaxační a sportovní služby. Každý takový podnik ale funguje jinak, poskytuje jiné služby a má své vlastní procesy a požadavky, které mohou být velice odlišné od požadavků ostatních podniků stejného druhu. Pokud danému podniku nevyhovuje žádný produkt aktuálně dostupný na trhu, lze přikročit k možnosti návrhu a tvorby vlastního systému, který bude přesně splňovat požadavky zadávajícího podnikatele.

Cílem této práce je analyzovat služby a potřeby sportovního centra a následně, za pomoci objektově-orientovaných metod a jazyka UML 2.0 navrhnout rezervační systém, který bude vyhovovat zadaným požadavkům. Tato aplikace by měla umožnit správu služeb, které dané sportovní centrum nabízí, dále by měla řešit správu permanentek, ze kterých si zákazník bude zarezervované služby hradit a v neposlední řadě by samozřejmě měla poskytovat zákazníkům sportovního centra pohodlný a jednoduchý způsob, jak si z domova zarezervovat požadované služby.

Struktura této práce se dělí na několik kapitol. Ihned po úvodu následuje kapitola č. 2, která se zabývá teoretickými a metodickými východisky objektově-orientovaného přístupu k modelování systémů. V této kapitole jsou rozebrány nejprve obecné základy vývoje systémů, následuje charakteristika některých starších objektově-orientovaných metodik, na které plynule navazuje kapitola, věnovaná čistě jazyku UML. Poté je popsán průběh analýzy systému a tvorby jednotlivých diagramů jazyka UML.

Následující kapitola č. 3 se zabývá analýzou výchozího stavu. V této kapitole jsou analyzovány potřeby a požadavky sportovního centra, spolu s motivací majitele sportovního centra pro tvorbu tohoto systému. Dále jsou zde popsány služby, které centrum poskytuje a shrnuty veškeré požadavky na systém.

Veškeré tyto požadavky jsou v následující kapitole s číslem 4 transformovány na návrh zadaného systému. Zde jsou popsány funkce, které systém bude splňovat a je zde rozebrána i jeho fyzická struktura.

Pátou kapitolu tvoří závěr, ve kterém budou vyhodnoceny výsledky práce.

2. Teoretická a metodická východiska objektivě orientované analýzy a návrhu informačního systému

V dnešní době se čím dál více klade důraz na využití moderních informačních technologií v podnikání. **Informační systémy a informační a komunikační technologie** (dále jen ICT) se stávají důležitou oporou podnikání v mnoha různých oborech. Proces zavádění těchto technologií je, dá se říci, nevyhnutelný. Zavádění ICT je jednak podnětem ke vzniku nových podnikatelských příležitostí a zároveň klíčovým bodem pro podporu a zefektivnění činností a procesů, probíhajících v různých typech organizací se širokým spektrem zaměření. (Tvrdíková, 2008)

Pojem informační systém (dále jen IS) je uživateli často používán intuitivně, ve významu veškerého **softwarového vybavení**, které daný podnik využívá. Jaká je tedy definice a z čeho se IS skládá? Podle Procházky a Klimeše (2009) je IS systém informačních technologií, dat a lidí, jehož cílem je efektivně podporovat informační a rozhodovací procesy na všech úrovních organizace. IS by měl v první řadě napomáhat podnikatelskému záměru, resp. pomoci naplnit podnikové cíle, což znamená nějakým způsobem usnadnit fungování či úplně automatizovat podnikové procesy.

Na světě je dnes mnoho odvětví podnikání, podniky mají různé podnikatelské záměry, jsou různě velké a v každém z nich probíhají specifické procesy. Proto i požadavky na funkce software se s jednotlivými společnostmi značně liší. Společnost má dnes možnost si koupit již vyvinutý software, určený k danému záměru (například různé ekonomické systémy) a spokojit se s nabízenou funkcionalitou, případně si nechat upravit konkrétní modul systému pro své vlastní potřeby. Druhou variantou je vývoj vlastního software od základů dle zadaných požadavků, ať už vlastním IT oddělením nebo externím dodavatelem.

Tímto se dostáváme k důležitému pojmu, kterým je projektování software neboli tzv. **Softwarové inženýrství**.

2.1 Softwarové inženýrství

„Softwarové inženýrství je inženýrská disciplína zabývající se praktickými problémy vývoje rozsáhlých softwarových systémů.“ (Vondrák, 2002, str. 6)

Z této definice podle Vondráka (2002) vyplývá, že vývoj softwarového systému zahrnuje celou řadu faktorů:

- technické aspekty, jako např. počítačovou infrastrukturu, softwarové vybavení;
- netechnické aspekty dané organizační strukturou společnosti, vyvíjející daný produkt;
- znalostmi z oblasti sběru požadavků, analýzy, návrhu, implementace, testování a nakonec také instalace u zákazníka;
- lidské zdroje, schopné úspěšně aplikovat výše uvedené znalosti;
- řízení spjaté s vývojem samostatného produktu, umožňující efektivní využití výše uvedených faktorů s cílem vytvořit požadovaný produkt.

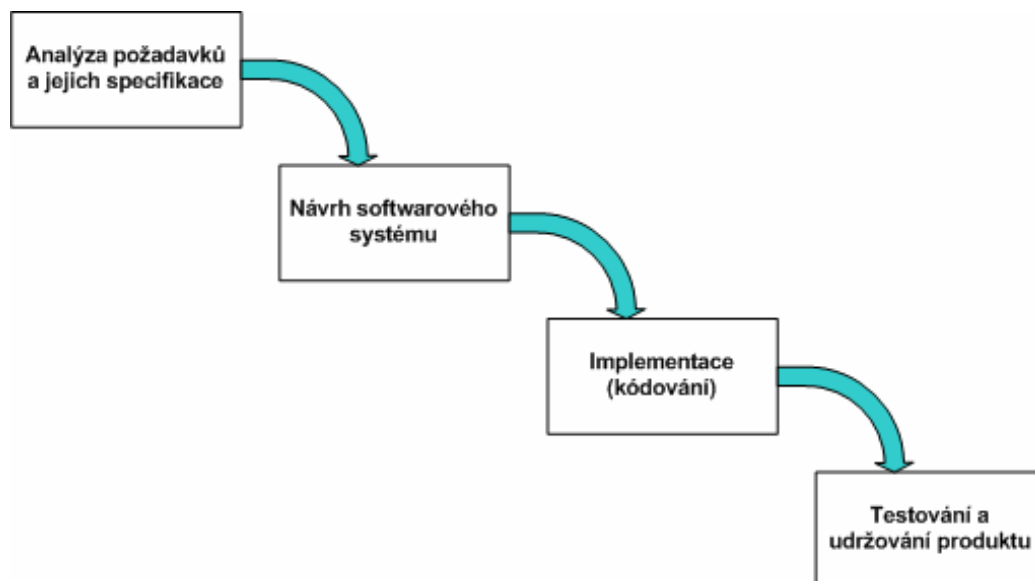
Vývoj softwarového produktu zahrnuje celou řadu činností a metod. Základem je tzv. **Softwarový proces**, což je postup činností, vedoucích k vytvoření produktu. Dle takto daného předpisu se následně definují **projekty** (projekt je v podstatě jedinečný proces), které jsou již vztaženy k jednotlivým zakázkám.

„Softwarový proces je po částech uspořádaná množina kroků, směřujících k vytvoření nebo úpravě softwarového díla.“ (Vondrák, 2002)

Obecně rozlišujeme dva základní přístupy k vývoji softwarových produktů. Těmi jsou tzv. **vodopádový přístup** a přístupy **iterativní** (neboli agilní).

2.1.1 Vodopádový přístup

Vodopádový model můžeme nalézt v různých modifikacích. Například Vondrák (2002) vychází z rozdělení životního cyklu softwarového díla na 4 fáze, kterými jsou analýza požadavků a jejich specifikace, návrh softwarového systému, implementace (kódování) a testování a udržování produktu.



Obrázek 2.1: Fáze životního cyklu vývoje softwarového díla ve vodopádovém modelu. (Vondrák, 2002)

Obecně je princip vodopádu takový, že následující skupina činností je bezprostředně navázána na tu předchozí a tudíž nemůže započít dříve, než je předchozí skupina činností ukončena. Tento základní model vývoje software má především tyto nedostatky:

- prodleva, která je mezi počáteční fází, resp. zadáním projektu a implementací spustitelného systému je příliš dlouhá,
- výsledek závisí na úplném a zcela přesném zadání požadavků, kladených na finální produkt,
- nelze odhalit výslednou kvalitu produktu danou splněním všech požadavků, dokud není výsledný produkt hotov.

Procházka a Klimeš (2009) uvádí, že základním rysem tohoto přístupu je tvorba detailních plánů ihned v úvodu, kdy ještě není spousta věcí jasných, nepočítá se v plánech s riziky a nečekanými událostmi. Požadavky uživatelů jsou dané a nemohou se měnit, jelikož integrace komponent a testování probíhá až v konečné fázi projektu.

2.1.2 Iterativní přístupy

V současné době vysoce sofistikovaných a složitých systémů je téměř nemožné nejprve specifikovat celé zadání a následně navrhnout řešení, vytvořit produkt a předat jej k užívání. Jediné možné řešení je přístup, postavený na iteracích, což znamená neustále se opakujících cyklech, umožňující postupné upřesňování zadání cestou rozšiřování z původní, hrubé formy do výsledné podoby. Systém se tak vyvíjí ve verzích, které jsou průběžně ověřovány zadavatelem a případně se v další iteraci pozmění dle potřeby. (Vondrák, 2002)

2.2 Objektově orientované metodiky a notace

Proces vývoje softwaru definuje při vývoji otázky *kdo*, *co*, *kdy* a *jak*. Tento proces vlastně realizuje uživatelské požadavky vytvořením softwarového vybavení. Důležitým nástrojem pro vývoj software je vizuální jazyk (notace), jako který je dnes primárně využíván nejnovější jazyk **UML (Unified modelling language)**, resp. **UML** ve verzi **2.0** i přesto, že existuje spousta dalších notací, které UML předcházely. Další důležitou částí je metodika vývoje, která definuje procesní část. Těchto metodik je rovněž spousta, s využitím UML je však nejčastěji spojována metodika **UP (Unified proces)**, i když ne nutně, jelikož neexistuje žádný obecný standard metodiky tvorby softwarového vybavení, který by jazyk UML doplňoval.

2.2.1 Některé starší metodiky a notace objektově-orientovaného modelování

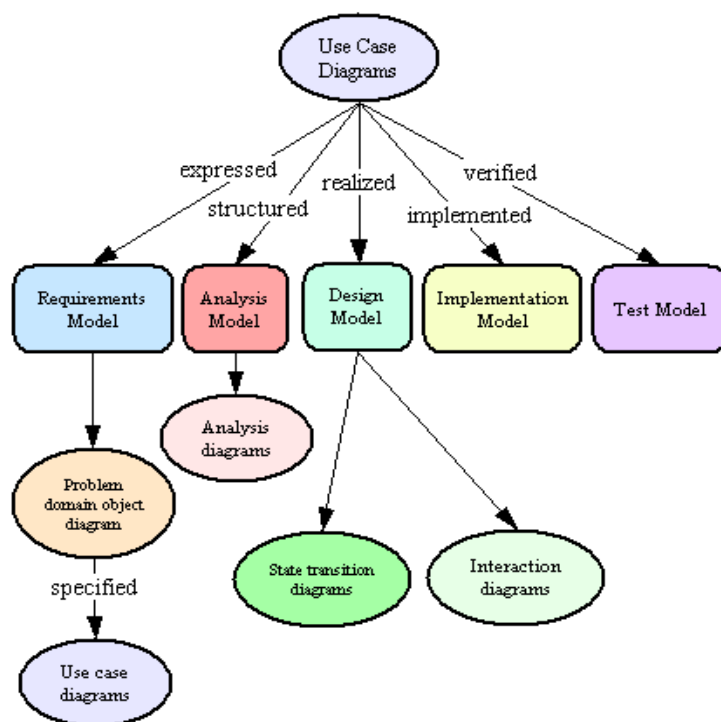
V průběhu 90. let bylo uvedeno na trh mnoho různých metodologií spolu s jejich vlastními sadami notací. Tři nejpopulárnější metody byly **OMT** (Rumbaugh), **OOAD** (Booch) a **OOSE** (Jacobson). Každá z těchto metod měla svoji vlastní hodnotu a důraz. OMT zdůrazňuje analýzu, OOAD návrh a OOSE analýzu chování.

Postupem času se metodiky začaly sbíhat, avšak měly dosud své vlastní notace. Používání odlišných notací působilo problémy na trhu, poněvadž jeden symbol znamenal různé věci pro různé lidi. Tato časová perioda je často nazývána válkou metod. Její konec nastal vznikem UML (Unified Modeling Language), které představuje sjednocení notací Booeche, Rumbaugh a objektových notací mnoha dalších metodologů. Při sjednocování notací použitých těmito objektově-orientovanými metodami poskytuje UML základ pro de

facto standard v oblasti objektově orientovaných analýz a návrhů založených na zkušenostech odborníků. (Objekty, 2012)

a) Metodika OOSE (Object-oriented software engineering - Ivar Jacobson)

Metodika, vyvinuta v roce 1992 Ivarem Jacobsonem a je jednou z předchůdců UML. Tato metodika kladla důraz převážně na analýzu chování a byla první, která využívala **use cases** (diagramy případů užití). Tento diagram zachycuje vztahy mezi aktéry (osoby nebo jiné objekty komunikující se systémem) a systémem. Podle Ivara Jacobsona hrají use cases dvě významné role. Jednak zachycují požadavky na funkcionalitu systému a jednak strukturují každý objektový model (Jacobson ve svých textech používá právě označení objekt, což zastupuje jak třídu, tak i její instance). Pro každý use case se vytvoří “samostatný” objektový diagram pouze s těmi objekty, které se účastní tohoto use case. Jeden objekt se může samozřejmě vyskytovat ve více objektových diagramech (modelovaných pro jednotlivé use cases). Podle toho určíme pole jeho odpovědnosti. Notace use case diagramů je obdobná jako u metodik zmíněných výše. Opět obsahuje aktéry use cases a vztahy komunikace mezi aktéry a use cases. Pro mapování use cases do objektových modelů se pak využívají dva typy notací, a to objektové diagramy a diagramy interakcí. Objektový diagram znázorňuje jednotlivé objekty a aktéry náležející danému use case. Objekty se znázorňují obdélníkem, aktéři jako panáček. (Objekty, 2012). Celkovou strukturu metodiky vyjadřuje následující obrázek.



Obrázek 2.2: Struktura metodiky OOSE (Object-oriented software engineering, 2012)

b) Metodika Booch – OOAD

Tato metodika se skládá ze čtyř hlavních aktivit a šesti notací. Pokrývá oblasti analýzy požadavků a analýzu reality, avšak hlavní důraz je kladen na design. Z metodiky vyzařuje i silný vztah přímo k programování.

Krok	Notace
logická struktura	diagram tříd (Class Diagram), objektový diagram (Object diagram)
fyzická struktura	diagram modulů (Module diagram), diagram procesorů (Process diagram)
dynamika tříd	stavový diagram (State Transition diagram)
dynamika instancí	diagram interakcí (Interaction Diagram)

Tabulka 2.1: Struktura metodiky Booch (Objekty, 2012)

Diagram tříd – Diagram tříd zachycuje všechny třídy objektů (viz třída), které se týkají modelovaného systému (tj. celou oblast zájmu řešení) a vztahy mezi těmito třídami.

Objektový diagram – Zachycuje objekty (instance tříd) a vztahy mezi nimi. Zachycuje tedy vlastně stav systému v určitém okamžiku.

Diagram modulů – Diagram modulů se používá pro zachycení rozmístění tříd a objektů do modulů ve fyzickém designu systému.

Stavový diagram – Stavový diagram zachycuje stavy objektu, kterými může projít v průběhu svého životního cyklu v systému.

Diagram interakcí – Diagram interakcí zachycuje posloupnosti interakcí (komunikace v určitém pořadí) mezi objekty (podobný sekvenčnímu diagramu v UML). (Objekty, 2012)

c) Metodika OMT (object modelling technique – James Rumbaugh)

Metodika OMT je vlastně reprezentantem oněch smíšených metodik, které za svůj základ vzaly ER diagram a rozšířily ho o nezbytné objektové rysy. V prvních verzích obsahovala OMT navíc ještě DFD diagramy, které však byly později vypuštěny.

Ve druhé verzi přebrala OMT Use Case modelování z metodiky OOSE (Jacobson), které je používáno především v analýze pro vymezení hranic systému a interakci systému s vnějším okolím. Na rozdíl od metodiky Booch se jako víceméně vhodné jeví i použití OMT ve fázi analýzy. Zkrácený životní cyklus je rozdělen do etap analýzy, systémového návrhu, objektového návrhu a implementace. OMT používá tři základní modely. Objektový (využití diagramů tříd vycházejícího z ERD), dynamický (zejména stavové diagramy vytvářené pro jednotlivé třídy) a funkční (právě tolik diskutované DFD). (Vývoj metodik a metod objektové analýzy a designu, 1999)

Objektový model obsahuje definice tříd a jejich vztahů společně s atributy a metodami (proto by byl asi lepší překlad názvu modelu jako model tříd, název objektový model může být terminologicky zavádějící). Objektový model zachycuje statickou strukturu systému.

Dynamický model zachycuje dynamiku objektů a změny jejich stavů. Zabývá se chováním objektů v čase a tokem zpráv a kontroly mezi objekty. Dynamický model zahrnuje stavové diagramy (STD - State Transition Diagram) pro každou třídu nebo pro důležité části návrhu. Dále pak diagramy interakcí.

Funkční model pak popisuje funkční závislosti systému, je podobný diagramu datových toků. Popisuje, co systém dělá (nezabývá se tím, jak to dělá). (Object modeling technique, 2012)

2.2.2 Jazyk UML a Unified process

Jazyk UML (Unified modeling language) je univerzální jazyk pro vizuální modelování systémů a byl navržen proto, aby spojil nejlepší existující postupy modelovacích technik a softwarového inženýrství. Sám o sobě nenabízí žádný druh metodiky modelování, samotný poskytuje pouze vizuální syntaxi, kterou můžeme využít při sestavování svých modelů. UML tedy umožňuje popisovat systémy pomocí slov a obrázků. Základem pro tuto schopnost je především variace rozmanitých grafů – ať už jsou to například use case diagramy se svým vyobrazením aktérů jakožto „panáčků“, tak třeba hojně užívané diagramy tříd. Ačkoli tyto diagramy nejsou nic nového (byly již využívány spolu se staršími metodikami objektově orientovaného modelování, zmíněnými v předchozí kapitole 2.2.1), unifikace modelovacích jazyků ve světovém měřítku přišla právě s UML, které bylo standardizováno **OMG (Object management group)**, což je mezinárodní organizace, podporující otevřené standardy pro objektově orientované aplikace.

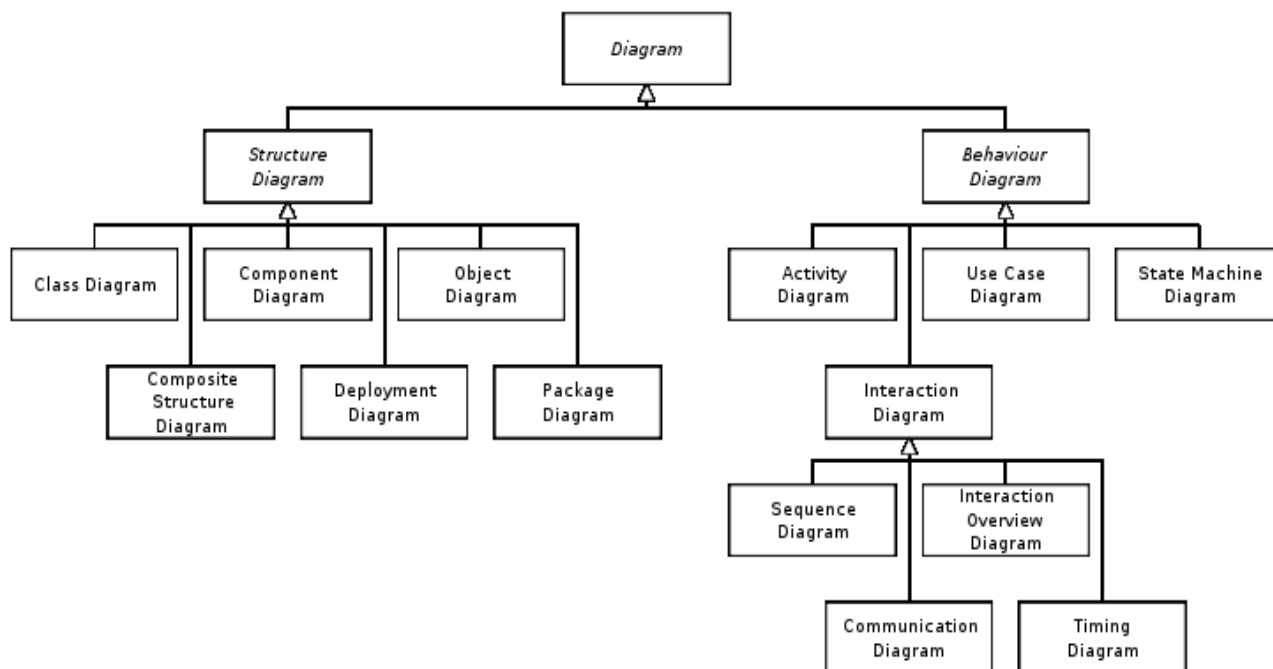
Existuje několik důvodů, proč využít UML jako modelovací jazyk:

- univerzalita terminologie a standardizace notace (diagramů) vede k významnému zjednodušení komunikace mezi všemi stranami, které se účastní vývoje,
- UML se rozrůstá spolu s rozrůstajícími se požadavky na modelování. Jelikož je UML velmi mocný modelovací jazyk, lze vývoj začít jak například velmi jednoduchým modelem, tak i modelem komplexním, plným všemožných detail,
- UML je vybudováno na velmi široce využívaných a prověřených přístupech. Jeho vývoj vychází primárně z problémů reálného světa a existujících modelovacích jazyků,
- je široce podporováno. (Arlow a Neustadt, 2007; Grässle aj., 2005)

Unified Process je již metodikou, která nám říká, jak pracovníky využít, jaké činnosti vykonat a jaké produkty vyrobit, aby se nám podařilo sestavit model funkčního softwarového systému. Jazyk UML není vázán na žádnou metodiku nebo životní cyklus. Lze jej použít společně se všemi možnými metodami a postupy. Metodiku Unified Process lze ovšem považovat za upřednostňovanou, jelikož využívá UML jako vlastní syntaxi pro vizuální modelování. Záměrem jazyka UML a metodiky UP byla od jejich vzniku podpora nejlepších postupů používaných v softwarovém inženýrství, vycházejících z ověřených zkušeností. (Arlow a Neustadt, 2007)

Diagramy jazyka UML

Ve verzi UML 2.0 existuje celkem 13 různých typů diagramů, které lze využít pro modelování systému.



Obrázek 2.3: Struktura diagramů v UML 2.0 (Hawtin, 2010)

Na obrázku 3 lze pozorovat, že diagramy jazyka UML lze rozdělit na dvě základní skupiny, kterými jsou **Strukturní diagramy** (Structure diagrams) a **Diagramy chování** (Behaviour diagrams).

Skupina strukturních diagramů vlastně popisuje fyzickou strukturu systému, slouží k popisu součástí modelovaného systému. Do této skupiny patří **diagram tříd** (class diagram), **diagram složené struktury** (Composite structure diagram), **diagram komponent** (Component diagram), **diagram nasazení** (Deployment diagram), **objektový diagram** (Object diagram), **diagram balíčků** (Package diagram).

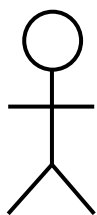
Skupina diagramů chování naopak slouží k popisu funkcionality systému. Modelují tedy, co daný systém musí dělat, popisují jeho chování. Mezi diagramy chování patří **diagram případů užití** (use case diagram), **diagram aktivit** (activity diagram), **diagram stavového automatu** (state machine diagram) a **diagramy interakce**, jejichž součástí jsou 4 diagramy, a to **sekvenční diagram** (sequence diagram), **diagram komunikace** (communication diagram), **diagram interakce** (interaction diagram) a **diagram interakce** (interaction overview diagram).

diagram), **diagram přehledu interakcí** (interaction overview diagram) a **diagram časování** (timing diagram).

a) Diagramy chování

Diagram případů užití

Diagramy případů užití se využívá k popisu chování systému z pohledu jeho uživatelů. Základními elementy těchto diagramů jsou **aktéři**, což jsou uživatelé systému a jednotlivé **případy užití**. Aktéři se obvykle v diagramech označují v podobě panáčků.



Zákazník

Obrázek 2.4: Aktér diagramu případů užití

Těmito aktéry jsou především osoby, které se systémem pracují. Jako aktéry je ale možné vnímat například i další systémy, které s daným systémem spolupracují a provádějí v něm určité úkony. Z pohledu na konkrétní osobu je jedno, kdo daný systém využívá, jedná se především o to, v jaké roli vystupuje. Například do rezervačního systému může vstupovat zákazník, který chce provést rezervaci nebo administrátor, který má povoleno podstatně více úkonů, které na systému může provádět. Dalším elementem jsou samotné **případy užití**. Tyto případy se znázorňují jako elipsy.

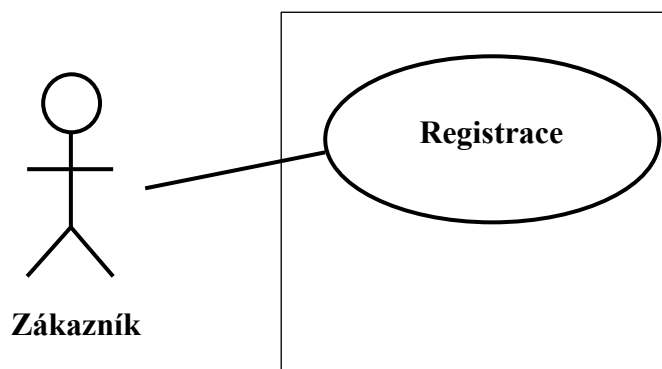


Obrázek 2.5: Znázornění případu užití

Případ užití popisuje chování, které probíhá mezi aktérem a daným systémem. Například chce-li zákazník využívat služby rezervačního systému, musí nejdříve *provést registraci*. Cokoli, co chce být uživatel v systému schopen dělat, musí být vyobrazeno jako případ užití. I když je myšlenka případů užití popisovat interakce mezi aktéry a systémem, je rovněž možné takto modelovat operace, které vykonává sám systém, jako například

automatické odstranění záznamů po určité době, nebo vytváření požadovaných statistik v určité periodě.

Pro tvorbu samotného diagramu je ještě nutné znázornit vztah mezi aktérem a případem užití. Toto se znázorňuje pomocí jednoduché vazby, která symbolizuje, že daný aktér může provádět tento případ užití. Tenká linka kolem případu užití naznačuje pomyslnou hranici systému. Lze tedy vidět, že aktér není součástí systému, ale vstupuje do něj zvenčí.



Obrázek 2.6: Znázornění případu užití

Jeden případ užití může být prováděn třeba i více aktéry. Například v rezervačním systému může zákazník provést rezervaci, ale zároveň jí může provést i administrátor. Administrátor může být vlastně takovým rozšířeným případem zákazníka, protože kromě zákaznických úkonů má povoleny ještě úkony administrační. Dále se v modelování případů užití využívají další typy vazeb, jako je například vazba **include**, která se využívá k vyjádření vztahu mezi dvěma případy užití. Tato vazba nám říká, že nějaký případ užití je součástí jiného. Ke znázornění se využívá přerušovaná čára se šipkou na konci, která směřuje k tomu případu užití, který je obsažen.

V modelování případů užití lze také využívat možnosti dědění. V takovém případě dědí odvozený případ užití veškeré funkce svého předka a následně je možné k němu přidat další funkce. (Arlow a Neustadt, 2007; Grässle aj., 2005)

Detailní informace o jednotlivých případech užití se značí do tabulek a jsou neopomenutelnou součástí modelování případů užití. Například podle Arlow a Neustadt, (2007) se do této tabulky zapisují následující informace:

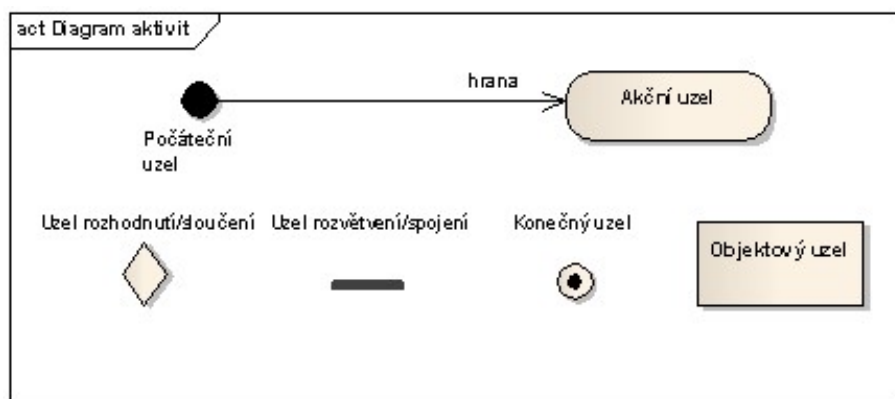
- ID případu užití, případně ID předka (pokud případ užití dědí), což jsou jednoznačné identifikátory pro každý případ užití,

- stručný popis činnosti, kterou případ užití vykonává, například „registruje zákazníka do systému a vytvoří mu účet“,
- hlavní aktéři, což jsou ti, kteří ve skutečnosti případ užití spouštějí,
- vedlejší aktéři, kteří mohou být v interakci s případem užití po jeho spuštění,
- vstupní podmínky, které určují, za jakých výchozích okolností může být případ užití spuštěn (například aby si zákazník mohl vytvořit rezervaci, musí být nejdříve přihlášen do systému),
- hlavní scénář, což je průběh (postup) vykonání daného případu užití, který vede k úspěšnému cíli bez jakýchkoli komplikací,
- alternativní scénáře, což jsou takové scénáře, které mohou nastat, pokud něco brání úspěšnému vykonání případu užití,
- výstupní podmínky, jimiž je výsledná změna, která v systému nastane po vykonání daného případu užití (například „systém provedl registraci nového zákazníka“).

Diagram aktivit

Diagram aktivit je možno použít pro popis procedurální logiky, modelování business procesů či různých pracovních postupů. Je možné jej využít rovněž k modelování jednotlivých případů užití jako posloupnosti akcí atd.

Diagram aktivit modeluje procesy prostřednictvím **uzlů**, které jsou propojeny **hranami**. Existují tři typy uzlů. **Akční uzly**, které reprezentují samostatné a v rámci aktivity již nedělitelné jednotky, **řídící uzly**, jejichž úkolem je řídit cestu uvnitř aktivity a **uzly objektové**, které zastupují objekty. Akční uzly jsou nejčastěji používány pro vyjádření aktivity, chování nebo operace, příkladem řídících uzlů jsou uzly **počáteční** nebo **koncové** či uzly **rozhodnutí** či **sloučení**. Uzel rozhodnutí má jednu vstupní hranu a několik hran výstupních. Daný výstup je pak zvolen podle toho, která ze zadaných podmínek je splněna. Uzel sloučení má naopak více vstupních hran a jednu výstupní, tudíž se používá pro sjednocení větví diagramu aktivit. Procesy, které diagram popisuje, mohou rovněž probíhat paralelně, k čemuž slouží uzly **rozvětvení** a **sloučení**. (Rejnková, 2009)



Obrázek 2.7: Prvky diagramu aktivit (Rejnková, 2009)

Diagram stavového automatu

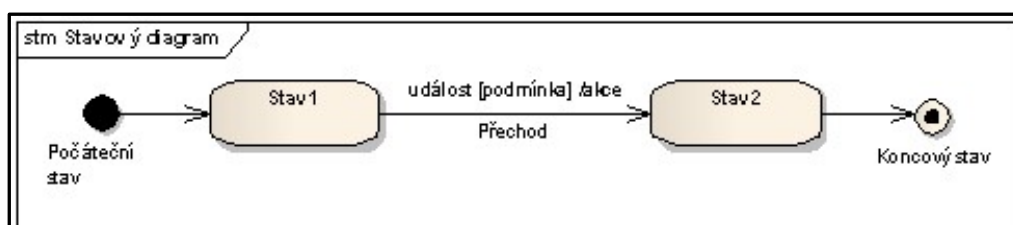
Stavové diagramy zachycují stavy objektů a přechody mezi těmito stavy. Využívají se především pro popis chování objektu napříč více případy užití.

Základními prvky stavového diagramu jsou **stavy**, **přechody** a **události**. Pokud se stavy nepohybují v cyklu, stavový diagram by měl obsahovat počáteční a koncový stav.

Stav je „*podmínkou nebo situací vzniklou během života objektu; v tomto stavu odpovídá určité podmínce, vykonává určitou aktivitu nebo čeká na určitou událost*“. (Rumbaugh, 1998 citovaný Arlow a Neustadt, 2007, str. 428)

Přechod je posunem z jednoho stavu do jiného v důsledku nějaké události. (Arlow a Neustadt, 2007)

Událost je „*specifikací určitého výskytu něčeho v čase a prostoru*“. (Rumbaugh, 1998 citovaný Arlow a Neustadt, 2007, str. 428)



Obrázek 2.8: Stavový diagram (Rejnková, 2009)

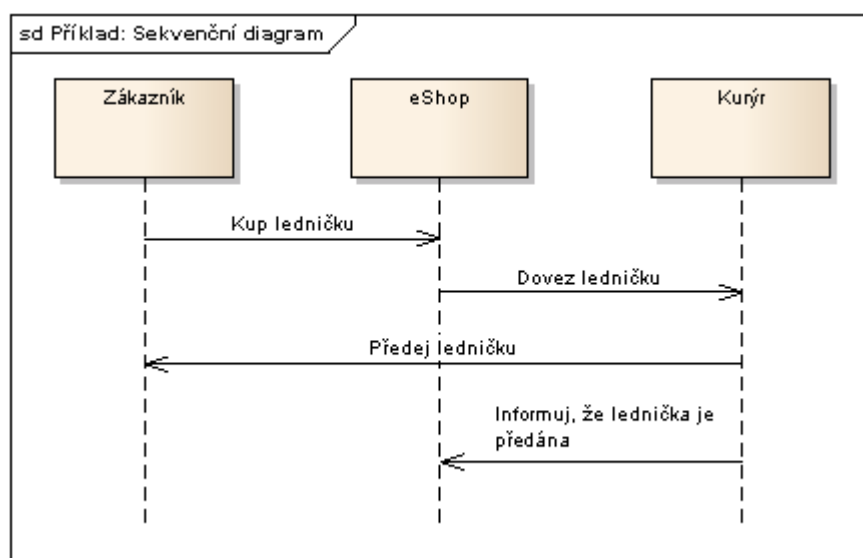
Sekvenční diagram

Sekvenční diagram patří do skupiny diagramů interakcí, které popisují spolupráci skupin objektů pro dosažení určitého chování. Nejčastěji vyobrazuje chování a spolupráci jednotlivých objektů v rámci jednoho případu užití.

Podle (Arlow a Neustadt, 2007, str. 253) sekvenční diagramy „znázorňují interakce mezi čárami života jako časově uspořádanou posloupnost událostí.“.

V sekvenčním diagramu jsou zprávy posílány mezi jednotlivými objekty, třídami nebo třeba i aktéry. Prvky, které mezi sebou v diagramu komunikují, se tedy označují souhrnně jako **klasifikátory**. **Čáry života** zastupují jednotlivé účastníky interakce (znázorňují způsob, jímž se instance daného klasifikátoru účastní interakce). Čas přitom běží shora dolů. Zprávy mezi účastníky jsou zobrazovány pomocí šipek. Tyto zprávy jsou vysílány shora dolů v pořadí, v jakém probíhají.

Jednoduchý příklad sekvenčního diagramu následuje na obrázku 9.



Obrázek 2.9: Sekvenční diagram (Rydval, 2011)

Diagram komunikace

Diagram komunikace je dalším ze skupiny diagramů interakcí. Na rozdíl od sekvenčního diagramu klade důraz na strukturální aspekty interakce umožněním libovolného rozložení instancí tříd, které jsou pospojovány a tato spojení jsou očíslována pro znázornění pořadí jejich komunikace.

Syntaxe diagramu je velmi podobná sekvenčnímu diagramu. Jednolivé objekty, aktéři či třídy ale nemají čáry života, jako je tomu u sekvenčního diagramu. (Rejnková, 2009)

Diagram přehledu interakcí

Tento diagram je absolutně novým diagramem, zavedeným v UML 2.0. Dalo by se říci, že je speciální variantou diagramu aktivit. Přínosem tohoto diagramu je to, že dokáže srozumitelně znázornit větvení a procesy, procházející více případy užití.

Jeho základními prvky jsou uzly a hrany, stejně jako u diagramu aktivit. Hlavním rozdílem je to, že uzly diagramu mohou zastupovat další interakce a ty mohou být znázorněny opět buď jako diagram nebo jen jako odkaz na něj.

Diagram časování

Poslední diagram ze skupiny diagramu interakcí a je dalším diagramem, zcela nově zařazeným v UML 2.0. Využívá se pro modelování systémů, pracujících v reálném čase. Jedná se o systémy, kde je čas rozhodujícím faktorem a každá událost musí proběhnout v přesně stanoveném časovém okně.

U diagramu časování je časová osa zobrazena rovně (zleva doprava), životní čáry a jejich stavy jsou svisle.

b) Strukturní diagramy

Diagram tříd

Třídy představují statický pohled na systém. Diagram tříd tedy znázorňuje typy objektů v systému a jejich vztahy, charakteristiky (atributy) a chování (metody). V systému je každý objekt instancí přesně jedné třídy. Všechny objekty stejné třídy musí mít stejnou množinu atributů, operací a relací. Atributy však obsahují (resp. mohou obsahovat) různé hodnoty (Arlow a Neustadt, 2007). Například v případě rezervačního systému může existovat Jan Novák, o němž si fitcentrum vede údaje v podobě jména, příjmení, e-mailu a kontaktu. Dále existuje Petr Novák, o němž si vede tytéž záznamy, je ale zřetelné, že hodnoty budou jiné. Oba tito lidé jsou z pohledu rezervačního systému instancemi jedné třídy – třídy *Zákazník*.

V diagramu tříd tedy obecně pracujeme s několika prvky. Základním jsou **třídy**. Třídy jsou abstrakcemi objektů, které mají některé stejné vlastnosti, vztahy či chování. Každá

třída obsahuje nějaké **atributy**. Ty reprezentují charakteristiky, které jsou důležité pro uživatele systému. Každá třída může provádět nějaké operace, označované jako **metody**. Dále mají mezi sebou třídy nějaké spojení, resp. **vztahy**. (Grässle aj. 2005)

Při tvorbě diagramu tříd můžeme využívat různé vztahy. Grässle aj. (2005) uvádí následující vztahy.

Generalizace – je určena ke znázornění vztahu mezi dvěma třídami, pokud je jedna třída nadřazena jiné třídě, a tato z ní dědí její vlastnosti. Vyjadřuje se prázdnou šipkou, směřující k nadřazené třídě.

Asociace – Asociace nám indikuje, že objekty jedné třídy jsou ve vztahu s objekty jiné třídy, kdy toto spojení může mít speciálně definovaný význam, například „skládá se z“ nebo „obsahuje“ apod. Vyjadřuje se obyčejnou čarou mezi dvěma třídami, která může být doplněna šipkou ve směru, který je relevantní k názvu a významu spojení.

Agregace – Toto spojení je speciální případ asociace. Říká, že objekty jedné třídy nemohou existovat samy o sobě, jsou podmíněny existencí objektů druhé třídy, která je s touto třídou ve spojení. Znázorňuje se čarou, na jejímž konci je tzv. „diamant“ (kosočtverec). Tento diamant leží na straně té třídy, jejíž existencí je podmíněna existence druhé třídy.

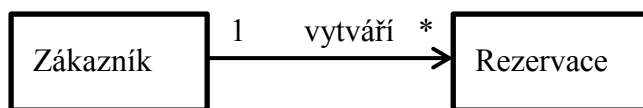
Dále můžeme v rámci vazeb mezi třídami vyjadřovat **multiplicitu**. Multiplicita vyjadřuje počet objektů, které je v dané asociační vazbě povolen. Základní hodnoty multiplicity jsou:

- 0...1, což je žádná a nebo právě jedna instance.
- 1 znamená právě jedna instance.
- *, případně 0...*, což znamená žádná až nekonečno instancí (hvězdička vyjadřuje nekonečno)
- 1...*, což vyjadřuje minimálně jednu až nekonečno instancí.

Multiplicitu je možné vyjádřit v libovolné podobě, resp. je možné u ní využít všechny možné kombinace hodnot, jako např. 2...4 nebo 6...*. Uvádí se u obou tříd v daném vztahu.

Následující obrázek říká, že třída zákazník je ve vztahu s třídou rezervací. Zákazník vytváří rezervace, což je slovně popsáno a šipka směřuje určeným směrem. Vyjádřená

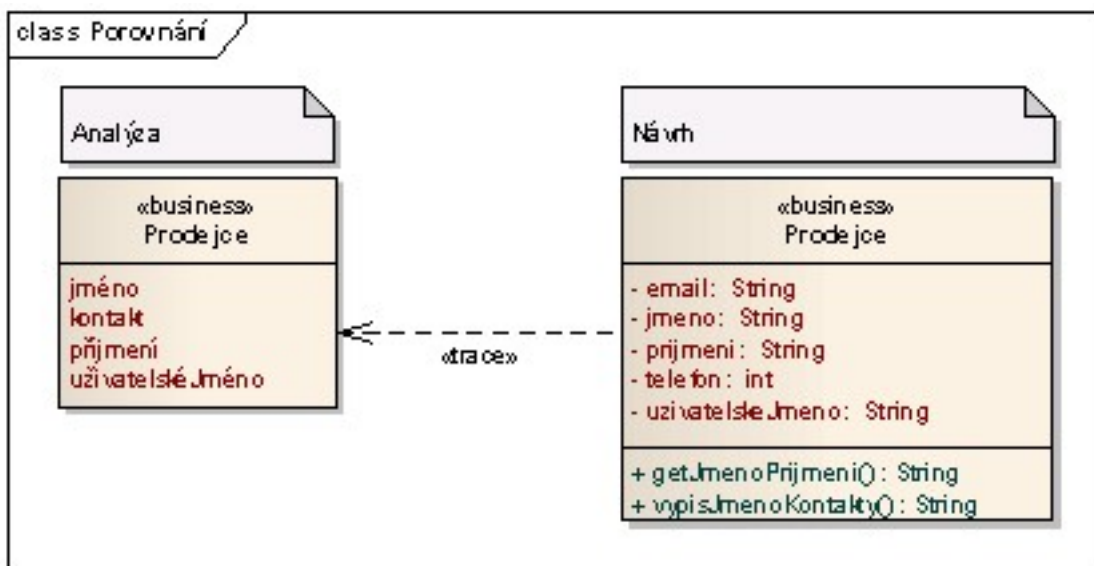
multiplicita říká, že každý zákazník může mít nekonečně mnoho rezervací (znázorněno * na straně rezervace), ovšem každá instance třídy rezervací může mít pouze jednoho zákazníka (1 na straně zákazníka).



Obrázek 2.10: Multiplicita ve vztahu tříd

Při konstruování diagramu tříd je nutné zamyslet se nad tím, zda jej potřebujeme pouze k vyjádření požadavků nebo zda potřebujeme získat detailní přehled o designu systému. Dle Arlow a Neustadt (2007) se rozlišují tři úrovně modelu tříd. První, tzv. **Analytický** (občas též konceptuální nebo doménový) je vytvářen ve fázi analýzy za účelem vyjádření požadavků na systém. U jednotlivých tříd se obvykle uvádí jen název třídy a názvy některých klíčových atributů a názvy základních klíčových metod. Pokud je diagram vytvářen pouze za účelem znázornění relací, není nutné vypisovat ani atributy, ani metody. **Návrhový** model (též designový) vychází z modelu analytického, zpřesňuje a zviditelňuje atributy a metody, přidává datové typy apod. Dále může do modelu přidávat také třídy, které obsluhují různé systémové události. **Implementační model** se zaměřuje na „grafické zobrazení implementovaného kódu“ (Buchalceková, 2007)

Na následujícím obrázku lze vidět srovnání, jak by mohla vypadat třída v rámci analytického modelu v porovnání s modelem návrhovým.



Obrázek 2.11: Porovnání analytické a návrhové třídy (Rejnková, 2009)

Objektový diagram

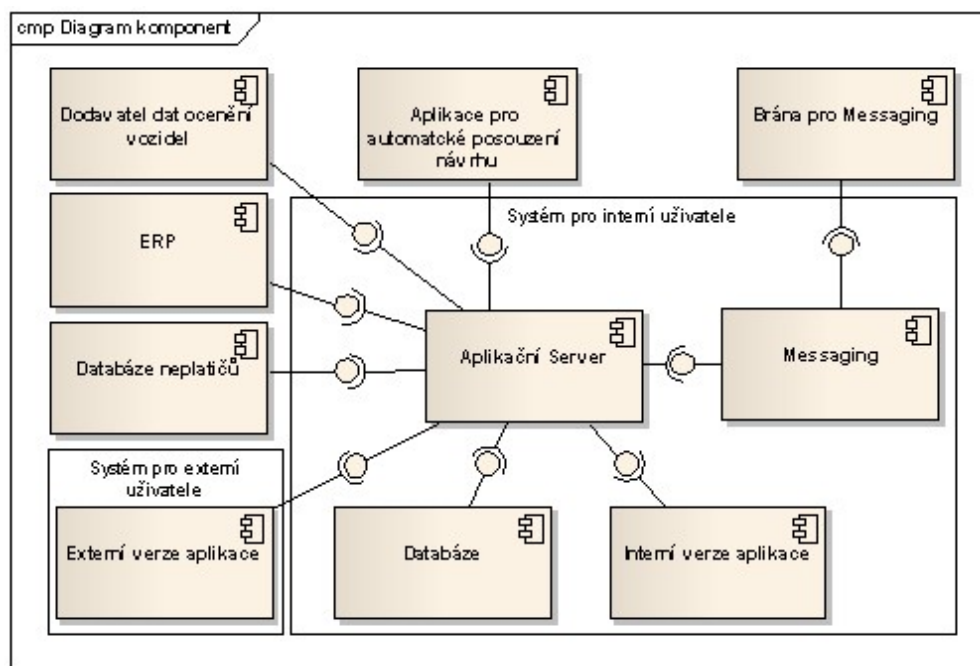
Objektový diagram je občas také nazýván diagramem instancí, jelikož zobrazuje instance tříd. Objektový diagram je vlastně takovým snímkem objektů a jejich vztahu v systému v určité chvíli. Používá se hlavně pro zobrazení vzájemně propojených objektů v situacích, kdy diagram tříd nestačí. Může být také chápán jako speciální případ diagramu tříd. Objektovým diagramem můžeme rovněž využít k modelování ukázek v počáteční fázi vývoje systému.

Notace objektového diagramu je velmi podobná diagramu tříd. Obvykle obsahuje pouze objekty a spojení mezi nimi. Atributy se uvádějí pouze, je-li to nutné, metody se u objektového diagramu neuvádějí vůbec. (Rejnková, 2009)

Diagram komponent

Diagram komponent ukazuje fyzický pohled na systém, strukturu softwarových komponent systému a závislosti mezi softwarovými komponentami. Diagram má podobu grafu komponent, propojených vzájemnými závislostmi. Tyto komponenty mohou obsahovat atributy a metody.

Nejllepší pro pochopení účelu diagramu komponent je jeho příklad na následujícím obrázku.



Obrázek 2.12: Diagram komponent v UML 2.0 (Rejnková 2009)

Diagram složené (vnitřní) struktury

Diagram, který byl nově zařazen do UML 2.0. Umožňuje znázornit interní strukturu komplexního prvku, jako je třída nebo případ užití a zobrazit spolupráci tohoto prvku s ostatními prvky v daném systému. Více o tomto diagramu a jeho konstrukci lze nalézt například na webové stránce orca.xf.cz/ooms.

Diagram nasazení

Pomocí tohoto diagramu se znázorňuje pohled na fyzické rozmístění systému. Hlavní prvky diagramu se nazývají uzly a jsou spojeny komunikačními cestami. Uzel je obecný název pro jakýkoli výpočetní zdroj. Rozlišujeme buďto uzly, které představují typ fyzické zařízení (hardware) a dále pak uzly, které reprezentují typ prostředí zpracovávání softwaru (např. webový server). Dále se zde vyskytují tzv. artefakty, což jsou fyzické soubory, díky kterým implementovaný systém pracuje na různých hardwarových uzlech.

Diagram balíčků

Tento diagram umožňuje sdružit elementy modelů UML (například třídy, nebo případy užití) do skupiny (tzv. balíčků) a mezi nimi znázornit závislosti. Balíčky se obvykle používají ke sdružení souvisejících tříd. Balíček vlastní svůj obsah a stejnou třídu tudíž nelze vložit do dvou balíčků, což ale nevyklučuje ji v jiném balíčku použít. Jeden z nich má však

vlastnické právo. Třídy v rámci jednoho balíčku musí mít vždy unikátní název. Sdružování tříd je možné provádět na základě architektury aplikace (například číselníky či uživatelská rozhraní). Lze je také seskupovat podle různých tematických oblastí (např. zákazník). Pro znázornění se využívá podobná metoda jako u tříd. Obdélník obsahuje název balíčku a uvnitř jsou vypsané jeho atributy. (Pastorčák, 2004; Rejnková, 2009; Arlow a Neustadt 2006)

Základy metodiky Unified process

Metodika UP je s UML nejčastěji spojovanou metodikou vývoje softwaru. Existuje rovněž její komerční verze s názvem RUP (Rational unified process), kterou dodává společnost IBM. Metodika RUP vlastně rozšiřuje metodiku UP, která je spíše otevřeným standardem a je mnohem obecnější.

Dle Arlow a Neustadt (2006) metodika UP obsahuje tři základní axiomy, kterými jsou:

- zásada řízení případem užití a rizikem,
- zásada soustředění se na architekturu,
- zásada iterace a přírůstku.

Dalo by se říct, že metodika UP se řídí požadavky a případy užití slouží k jejich zachycení a modelování. Soustředí se rovněž na rizika spojená s tvorbou softwaru a posuzuje jeho stavbu na základě analýzy těchto možných rizik. Druhý bod říká, že metodika UP je založena na návrhu a postupném vývoji robustní architektury systému. Architektura systému popisuje jak strategické aspekty možnosti rozkladu systému na komponenty, tak i způsob, jakým se tyto komponenty vzájemně ovlivňují a jakým jsou nasazeny do příslušného hardwaru. Celá metodika je založena na iterativním vývoji, což znamená, že projekt je rozdělen na menší podprojekty, jimiž je projekt vytvářen po malých částech. Tvorba každé této malé části systému prochází několika fázemi, kterými jsou:

- požadavky,
- analýza,
- návrh,
- implementace,
- testování.

2.3 Analýza a návrh systému pomocí UML

2.3.1 Specifikace požadavků (Inženýrství požadavků)

Termín Inženýrství požadavků popisuje různé aktivity, které souvisí se zjišťováním a dokumentováním požadavků v procesu tvorby informačního systému. Tento proces zajišťuje tzv. Systémový analytik. Obecně se rozlišují dva druhy požadavků. Funkční požadavky formulují, jaké činnosti by měl systém vykonávat z pohledu zákazníka (například registrace zákazníka, přihlášení, vytvoření rezervace). Na druhé straně nefunkční požadavky se již zabývají omezujícími podmínkami implementace systému (například v jakém programovacím jazyce bude systém napsán). Úkolem systémového analytika je ovšem prioritně se zabývat funkčními požadavky a oprostit se od implementačních technologií (Arlow a Neustadt, 2007).

Význam požadavků nejlépe vystihuje následující věta:

„A requirement describe what a system is supposed to do, not HOW the system is supposed to do it“ (George aj. 2007)

Z této definice vyplývá, že na požadavky je třeba nahlížet z pohledu toho, co uživatel od systému potřebuje. V této fázi nezáleží, jak systém bude uvnitř fungovat, v jakém jazyce bude naprogramován apod. Jaké akce bude systém vykonávat a k čemu bude sloužit lze zjistit z různých zdrojů, které se systémem souvisí. Těmi mohou být například:

- zákazníci, přímí uživatelé systému,
- administrátoři, správci, údržba systému,
- další systémy, se kterými bude systém v interakci,
- právní a regulační omezení, obchodní cíle. (Arlow a Neustadt, 2007)

Sběrem požadavků se zabývá Systémový analytik. Pro shromáždění požadavků existuje spousta různých metod, od tradičních rozhovorů až po nové moderní metody jako např. JAD (Joint application design). Aby byl proces shromažďování požadavků efektivní a výsledky byly co nejpoužitelnější pro tvorbu systému, je potřeba, aby se systémový analytik držel jistých „zásad“, kterými jsou, podle George aj. (2007):

- Nebát se a ptát se do detailů, ptát se na všechno ohledně současných potřeb i na budoucí vize systému. V případě rezervačního systému pro sportovní centrum by to mohly být otázky jako „Plánujete do budoucna možnost založit další pobočku?“.

- Být nestranný. Systémový analytik má za úkol najít nejlepší řešení problému. Musí zvážit všechny otázky z každého úhlu pohledu a co nejefektivněji je propojit.
- Odpoutat se od omezení. Analytik by neměl přijmout prohlášení jako že „se něco vždycky dělalo takhle, tak se v tom bude pokračovat i nadále“. Jistě existuje důvod, proč tato tradice vznikla, ale organizace i její prostředí se v čase vyvíjí a mění a postupy v minulosti efektivní se mohou obrátit pouze v nesmyslný zvyk.
- Dávat pozor na detaily. Žádná informace nesmí rozporovat jinou informaci.
- Být kreativní. Analýza je kreativní proces a úkolem systémového analytika je vcítit se do potřeb zákazníka a přemýšlet nad požadavky z jeho pohledu. Práce na obdobném projektu v minulosti by neměla být východiskem současného projektu. Východiskem by měla být představa současného zákazníka.

Dle George aj. (2007) se metody sběru požadavků dělí do dvou základních kategorií. Těmi jsou tradiční metody, a metody moderní.

a) Tradiční metody

Sběr informací je jádrem systémové analýzy. Na začátku je potřeba shromáždit informace o systému, který zákazník aktuálně používá. Je důležité zjistit, co by uživatelé systému rádi vylepšili. Jedna z nejlepších cest je tedy provádět rozhovory s lidmi z různých částí organizace, kterých se případná změna systému dotkne. Tradiční metody tak představují cestu, jak získat informace přímo ze zdroje. Typickými tradičními metodami pro sběr požadavků jsou rozhovor, pozorování a analyzování dokumentace.

Rozhovor. Jedná se o nejzákladnější metodou pro sběr požadavků. Na počátku projektu tráví systémový analytik spoustu času prováděním rozhovorů s lidmi. Vyptává se na jejich práci, informace, které k ní využívají, od jiných se získávají informace o organizačních pravidlech, normách a očekávání manažerů. Během rozhovoru jsou zaznamenávány veškerá fakta, názory a spekulace, zároveň se bedlivě sleduje řeč těla a emoce.

Pozorování. Jedná se o další tradiční metodou sběru informací pro tvorbu systému. Lidé často při osobním rozhovoru nejsou schopni podat relevantní a důvěryhodné odpovědi na otázky, týkající se jejich pracovní náplně. Často podávají informace zkreslené a odpovídají tak, jak si myslí, že se od nich očekává. Osobní rozhovor může být tedy doplněno a jeho výsledky mohou být podpořeny pozorováním uživatelů systému v jejich pracovním procesu. Nevýhodou může být změna chování z důvodu uvědomění si, že je dotyčný někým

pozorován. Takovýto člověk může být nervózní a dělat více chyb než obvykle, nebo se naopak snažit dodržovat veškeré pracovní postupy až příliš důsledně.

Analýza dokumentace. Tato metoda spočívá ve čtení různých typů firemních dokumentací. Těmito dokumentacemi mohou být například různé dokumenty, které popisují pozice jednotlivých zaměstnanců, dokumenty organizační struktury (např. grafy), vnitropodnikové směrnice a nařízení, firemní korespondence, katalogy, portfolio nebo také dokumentace současného informačního systému

c) Moderní metody

I když rozhovor, pozorování a analýzu dokumentů nazýváme tradičními metodami pro sběr uživatelských požadavků, tyto metody jsou stále analytiky k tomuto účelu hojně využívány. Podle George aj. (2007) existuje ale několik dalších, tzv. moderních metod, které jsou schopny efektivně podporovat sběr požadavků na systém, zatímco dokáží výrazně šetřit čas systémovým analytikům. Těmito metodami jsou hlavně metoda JAD (Joint application design) a metoda prototypování při sběru požadavků na systém.

JAD (Joint Application Design). Tato metoda odstartovala již na konci 70. Let v IBM a jejím význam spočíval ve shromáždění klíčových uživatelů, systémových analytiků a manažeru. Hlavní filozofií metody JAD bylo mít hlavní klíčové účastníky projektu pohromadě a sledovat oblasti, ve kterých se shodují a oblasti, kde naopak nastává střet názorů. Schůze jsou obvykle lokalizovány do míst absolutně mimo pracoviště zúčastněných, aby nebyli rozptylováni a vyrušováni a mohli se soustředit pouze na systémovou analýzu. Metoda JAD může trvat 4 hodiny, nebo klidně i celý týden a může být složena z několika různých schůzí. Také z toho důvodu je velmi nákladná na finanční zdroje. Nejdražší je samozřejmě čas zúčastněných, další velký podíl mají náklady spojené s místem konání (náklady na dopravu, ubytování atd.).

Prototypování. Jedná se o opakující se proces, kdy analytici budují určitou „základní“ verzi systému pro komunikaci s uživatelem. Nemusí se ani jednat o ryze funkční systém. Je možné prototypovat i formou „falešného“, pouze grafického návrhu uživatelského rozhraní. Jde v zásadě o to, aby si uživatel odzkoušel a zhodnotil, jaké v něm chybí či co by bylo třeba změnit nebo vylepšit. Může se takovouto cestou rovněž vyjádřit ke grafickému designu dané aplikace. Analytik posléze může změnit návrh a opět jej ukázat zákazníkovi. Prototypováním rovněž mohou vznikat nové případy užití pro daný systém.

2.3.2 Tvorba případů užití

Modelování případů užití je způsob, jak popsat požadavky na systém. To, jakým způsobem se znázorňují jednotlivé prvky v diagramech, již bylo popsáno částečně v kapitole 2.2.2. Na počátku modelování případů užití je důležité si určit hranice systému. Těmito hranicemi se určí, co je součástí systému, a co jeho součástí není. Je důležité se zamýšlet nad tím, kdo systém bude využívat, kdo nebo co s tímto systémem bude spolupracovat a co přesně systém bude vykonávat. Ve chvíli, kdy nalezneme všechny aktéry systému a všechny případy užití, stává se tato hranice mnohem zřetelnější.

Případem užití je vlastně jakákoli funkce systému, kterou využívá některý z aktérů systému. Pro zpřesnění případů užití je možné využít i tvorbu prototypů (klidně i nefunkčních), kterými si lze vizualizovat potenciální uživatelské prostředí systému a lze tedy potřebné funkce systému lépe odvodit či vůbec si je lépe představit

Samotné případy užití je poměrně dobré si rozdělit do několika skupin podle toho, jaké části systému se týkají a co vyjadřují, jelikož často desítky případů užití v jednom diagramu se mohou stát poněkud nepřehlednými.

Po namodelování veškerých případů užití a znázornění do diagramů se obvykle vytváří tzv. **RTM matice**, resp. matice sledovanosti (či trasovatelnosti) požadavků. Tato matice umožňuje obecně modelovat a následně dohledávat vztahy mezi jednotlivými projektovými výstupy, zde tedy sledovat vztahy mezi uživatelskými požadavky a případy užití. Zápis ve formě matice umožní rychle a přehledně zjistit, zda jsou všechny požadavky uživatele zastoupeny alespoň jedním případem užití. Tak je možné vyloučit, že se na některý z uživatelských požadavků zapomnělo.

případy užití	případy užití 1	případ užití 2	případ užití 3
požadavky			
požadavek 1			
požadavek 2			

Tabulka 2.2: Matice sledovanosti požadavků

2.3.3 Tvorba tříd

Jak již bylo zmíněno v kapitole 2.2.2, můžeme rozlišovat analytický, návrhový a implementační model tříd. Analytický model tříd zahrnuje pouze základní třídy a její základní atributy, nezaobírá se většinou ani metodami, ani datovými typy atributů. Je zkrátka určen ke komunikaci mezi zákazníkem a analytikem, aby zákazník lépe porozuměl fyzické struktuře systému. Návrhová třída je již poněkud detailnější. Ať už se jedná o analytické či návrhové třídy, existují různé metody, jak tyto třídy na základě předchozí analýzy nalézt.

Arlow a Neustadt (2006) uvádí, že třídy můžeme hledat například na základě **analýzy podstatných jmen a sloves**, kdy prohledáváme text (nebo přímo popisy diagram případů užití) a hledáme podstatná jména nebo fráze tvořené podstatnými jmény, které vyjadřují třídy, případně atributy a slovesné fráze, které obvykle vyjadřují metody (operace) daných tříd. Tato metoda se s úspěchem provádí již dlouhá léta, má ovšem také své nevýhody. Výskyt různých synonym a homonym může vést ke vzniku nesprávných a nežádoucích tříd. Postup této metody je velmi jednoduchý. Nejdůležitější je shromáždit si maximální počet důležitých informací. Vhodnými zdroji mohou být například slovníček pojmů projektu, specifikace doplňkových požadavků nebo již dříve zmíněné případy užití. V následující fázi se analyzují získané materiály, jednoduše zvýrazněním podstatných jmen (či spojení několika podstatných jmen), sloves a slovesných frází.

Další metodou je dle Arlow a Neustadt (2006) tzv. **Metoda štítků CRC**. Při této metodě zapojujeme do hledání tříd také uživatele systému. Tato metoda by se měla používat, s výjimkou velmi jednoduchých systémů, společně s předchozí metodou (analýzou podstatných jmen a sloves). Postup této metody je poměrně triviální. Na počátku se sejdou analytici, zainteresované osoby a další různí potřební odborníci. Ti pak společně provozují takový “brainstorming”. Členové týmu pak pojmenovávají různé “předměty”, které figurují v jejich obchodní doméně. Každý tento předmět se zapíše na samolepící lísteček, jakožto kandidát na třídu nebo nějaký její atribut. Tento lísteček se poté nalepí na tabuli. Na lístečky se poté dopisují odpovědnosti, které předmět může mít a také další předměty, které s daným předmětem mohou spolupracovat. Poté se sejdou systémoví analytici a odborníci v jednotlivých doménách, které společně rozhodnou, jak transformovat uvedené předměty na jednotlivé třídy a jejich atributy.

Nakonec ještě existuje několik dalších metod, které se souhrnně nazývají hledáním **tříd pomocí stereotypů metodiky RUP**. Více informací o těchto metodách hledání tříd uvádí Arlow a Neustadt (2006).

Poté, co si identifikujeme jednotlivé třídy a jejich atributy, v návrhové fázi se již do diagramu tříd zapisují také datové typy daných atributů. Existují různé datové typy, které můžeme přiřadit jednotlivým atributům. Přiřazení datových typů je závislé na tom, co bude daný atribut vyjadřovat a zároveň každý programovací jazyk má vlastní sadu datových typů. Například pokud máme atribut, který bude vyjádřen psaným textem (může to být například jméno zákazníka, příjmení nebo klidně i delší popis), využije se datového typu **String**, který vyjadřuje řetězec znaků. Pokud hodnota bude nabývat celých čísel, použije se datový typ **int** (integer), který nabývá celočíselných hodnot (například 1,2,3 ale i 12345 atd.) Pokud je potřeba zapisovat údaj ve formě data, využije se k tomu pravděpodobně datový typ **Date**. K zapsání času např. **Time** apod.

Když už jsou v diagramu zapsány atributy a jejich datové typy, případně metody, je nutné identifikovat spojení mezi jednotlivými třídami. Spojení, resp. vazby mohou být různého typu, tyto typy vazeb a jejich využití jsou uvedeny v popisu třídních diagram v kapitole 2.2.2.

2.3.4 Diagramy aktivit

Diagramy aktivit se využívají k upřesňování jednotlivých případů užití, resp. jejich grafickému modelování. Modelují se tak scénáře případů užití, jako posloupnost akcí. Využitelné jsou převážně v raných fázích analýzy, jakožto komunikační prostředek mezi uživateli system a analytiky. V diagramech aktivit je počáteční a koncový uzel, jednotlivé uzly mezi nimi znázorňují aktivity, které jsou postupně vykonávány v rámci jednoho případu užití. Diagramy aktivit je vhodné využít v případě například delších scénářů, případně většího množství alternativních scénářů. V takovém případě se v něm znázorňuje větvení na různé scénáře pomocí speciálního typu uzlu (viz popis diagramu aktivit v kapitole 2.2.2 o diagramech jazyka UML). (Arlow a Neustadt, 2007)

2.3.5 Realizace případů užití – Diagramy interakcí

Diagramy interakce jsou využívány k modelování různých interakcí mezi jednotlivými objekty (třídami). Vyjadřují tedy, jak mezi sebou různé třídy komunikují v rámci jednoho případu užití.

Do skupiny diagram interakcí patří především sekvenční diagram, který zdůrazňuje časově orientovanou posloupnost zpráv, které si jednotlivé objekty tříd předávají mezi sebou v rámci jednoho případu užití (nebo i více, pokud spolu dané případy užití nějakým způsobem úzce souvisí). Tyto objekty komunikují i s aktéry systému, kteří jsou většinou spouštěčem případu užití. Aktéři komunikují s jednotlivými objekty přes nějaké uživatelské rozhraní (GUI – graphical user interface).

Dalším diagramem je komunikační diagram, který je velmi podobný sekvenčnímu. Zabývá se rovněž vyjádřením komunikace mezi objekty tříd, jejich podoba je však poněkud jiná, nejsou tolik orientovány na vyjádření časové posloupnosti.

Diagram zjednodušené interakce ukazuje složité chování a to tak, že zahrnuje množinu jednodušších interakcí. Je to vlastně zvláštní případ diagram aktivit, jehož uzly se odkazují na jiné diagramy.

Posledním diagramem této skupiny je diagram časování, který slouží je zdůraznění významu toku času. (Arlow a Neustadt, 2007)

2.3.6 Diagramy stavového automatu

Stavové automaty lze využívat v mnoha fázích analýzy a návrhu. Mohou se používat k modelování životního cyklu tříd, kde stavy hrají významnou roli, ve fázi návrhu k modelování souběžnosti, nebo dokonce ještě ve fázi sběru požadavků k pochopení složitých případů užití. Základní otázkou je, zda tvorba stavového automatu bude mít užitečný význam. Pokud stavový automat pomůže pochopit složitější životní cyklus či chování, pak má jeho konstrukce smysl. Pokud daný objekt v průběhu svého životního cyklu může procházet různými stavy v důsledku nějakých akcí, pak je konstrukce diagramu stavového automatu vhodná. Samotná konstrukce diagramu je poměrně jednoduchá, je třeba si ujasnit stavy objektu a akce, díky kterým daný objekt jednotlivými stavy prochází. Každý stavový automat má počáteční a koncový uzel, pokud se jeho stavy ovšem nepohybují v cyklu. Dále jednotlivé stavy jsou znázorněny pomocí zaokrouhlených obdélníků a přechody mezi nimi pomocí šipek.

2.4 Shrnutí

V této kapitole byly popsány teoretické a metodické základy dané problematiky, které budou následně využity v kapitole č. 3 a kapitole č. 4. Veškeré informace, které byly v kapitole o teoretických a metodických základech, byly již vybrány s ohledem na potřeby praktické části této práce. Z veškerých popsaných diagramů jazyka UML budou vybrány ty diagramy, které jsou vhodné pro tvorbu analýzy a návrhu systému, k tomuto účelu bude rovněž zvolen vhodný pracovní postup.

3. Analýza současného stavu

Návrh rezervačního systému vychází z požadavku Sportovního centra, které se nachází v Ostravě.

Pro zanalyzování služeb a konkrétních potřeb sportcentra byla použita metoda osobního rozhovoru se zákazníkem (majitelkou sportcentra). Po předchozí přípravě struktury bylo položeno dotazované osobě několik otázek, týkajících se současného řešení rezervací a požadavků na nový rezervační systém. Jako doplňková metoda byla využita analýza dokumentace.

Sportcentrum si zadalo jako požadavek tvorbu online rezervačního systému pro několik druhů služeb, kterými jsou:

Spinning

H.E.A.T. program

Služby aerobního sálu

Infrasauna

Masážní studio

V současné době probíhá rezervace služeb sportcentra především telefonicky, případně e-mailem a osobní žádostí přímo na pobočce. Rezervace jsou na jméno zákazníka a jsou hrazeny obvykle v hotovosti na místě, případně ze zakoupené permanentky, které jsou provozovány formou čipových karet. Veškeré tyto rezervace spravuje zaměstnanec sportcentra. **Výhodou** tohoto systému jsou relativně nízké náklady na provoz, jelikož zaměstnanec, který se o veškerý chod sportcentra stará musí majitel platit v obou případech. Za další výhodu by se dala označit jednoduchost užívání z pohledu zákazníka. Zatelefonovat či napsat e-mail již dnes umí snad každý. Dalo by se taky říct, že ve prospěch tohoto systému rezervací hraje i jeho minimální závislost na nějaké hardwarové či softwarové základně. Na druhou stranu je však možné tyto zdánlivě výhodné stránky současného stavu přehodnotit a obrátit je v poměrně velké **nevýhody**. Například současný zaměstnanec může být množstvím rezervací příliš vytížen, což následně může vést k nespokojenosti zákazníků. Telefonickou objednávku může provést v jednu chvíli pouze jeden zákazník, a pokud zaměstnanec sportcentra aktuálně tuto telefonickou objednávku přijímá, pak nemůže obsluhovat zákazníky

u pultu a podobně. V dalším případě je sice pravda, že učení používání nového rezervačního systému může zákazníkům zpočátku poplést hlavu, na druhou stranu je pohodlnější provést jednoduše rezervaci několika kliknutími, než se snažit dovolat na přetíženou recepci. I přes zdánlivě výhodnou nezávislost na hardwaru a softwaru, tento systém neumožňuje žádnou pohodlnou manipulaci s daty. To se může týkat např. vyhledávání zákazníků, tvorby nějakých sestav nebo například problému náhlého zrušení hodiny, kdy nakonec zaměstnanci sportcentra nezbyde nic, než obvolat či obeslat e-mailem všechny zarezervované osoby.

Ovšem hlavním důvodem požadavku na rezervační systém byly dle slov majitelky nekontrolované a nezávazné rezervace, hlavně tedy v rámci programu H.E.A.T., kdy zákazníci nedocházeli na rezervované lekce. Tím bylo zabráno místo ostatním zájemcům a následně, vzhledem k platbám prováděným v hotovosti na místě, byl zaznamenáván ušlý zisk a fitcentru se prodlžovala doba splácení zakoupených strojů a dalších cvičebních pomůcek. Majitelka tedy vyjádřila potřebu zavedení kontrolovaných rezervací online, které by bylo možné provést pouze registrovaným zákazníkem a jejichž platby by bylo možno realizovat pomocí dobitého kreditu (zakoupené permanentky). Tím by se naskytla možnost strhnout propadlé rezervace z účtu zákazníka a eliminovala by se ztráta sportcentra.

Dle zadání majitele sportcentra by měl rezervační systém fungovat tak, aby řešil a automatizoval všechny procesy, které v tomto centru probíhají a které souvisí s poskytováním nabízených služeb zákazníkům.

Součástí požadavků na systém bylo, jak již vyplývá z předchozího textu, rovněž zahrnutí principů poskytování slev a permanentních vstupenek. Sportcentrum poskytuje následující základní typy permanentek.

Vstupové permanentky, které se vždy vztahují k nějaké službě a zakupují se na počet vstupů.

Kreditní vstupné, kdy zákazník může jednoduše čerpat z předplacené částky.

Časové permanentky, vztahující se většinou pouze k posilovně.

Aktuálně sportcentrum vydává čipové karty, na kterých jsou načteny informace o předplacené částce.

3.1 Analýza služeb sportcentra

Jak již bylo zmíněno výše, sportcentrum požaduje rezervační systém na celkem pět typů konkrétních služeb, kterými jsou Spinning, H.E.A.T., Aerobní sál, Infrasauna a masážní studio. Tyto služby je možno rozdělit do dvou základních typů:

Služby jednorázové

Služby celodenní

3.1.1 Služby jednorázové

Do této kategorie spadají služby spinningu a aerobního sálu. Tyto služby mají zadány konkrétní časy, ve kterých jsou pořádány a na které je možné přihlásit větší počet zákazníků.

Součástí sportcentra je samostatný spinningový a aerobní sál.

Spinning je pořádán systémem předem vypsaných hodin, na které je možné přihlásit počet zákazníků, který je roven počtu funkčních spinningových kol. Na stejné bázi funguje i služba H.E.A.T.

V aerobním sále se provozuje několik typů služeb, kterými jsou například aerobik, pilates či cvičení na balónkách. Na všechny tyto typy cvičení jsou vypsané rovněž předem nadefinované hodiny, na které je možné přihlásit počet lidí, který si zvolí pracovník sportcentra (provozovatel lekce), a který odpovídá kapacitě cvičebny.

U všech těchto služeb je velice důležité vědět, který instruktor vede danou hodinu, jelikož se od toho odvíjí preference zákazníků. Každý instruktor organizuje hodinu individuálním způsobem, je tedy nutné, aby zákazník při rezervaci věděl, kdo hodinu povede. Další důležitou informací pro zákazníka je místo výkonu, tedy cvičiště, na kterém bude služba vykonána.

3.1.2 Služby celodenní

Služby na úrovni celodenních rezervací fungují tak, že jsou celý den přístupné (obvykle po celou dobu otevírací doby sportcentra) a je u nich zadán minimální časový úsek, na který se lze rezervovat. Například infrasaunu je možné využívat po minutách, lze na ni tedy provést rezervaci na několik po sobě jdoucích minut, a to kdykoli během dne. Dále jsou služby, které obsahují tzv. **Podslužby**.

Službou s podslužbami jsou služby masážního salónu, kdy se zákazník jednak hlásí ke konkrétnímu pracovníkovi, a následně si vybírá podtyp služby (různé druhy masáží), které mají jiné doby trvání i ceny.

V případě infrasauny neexistuje žádný instruktor, který by službu vedl, infrasaunu zkrátka zákazník využívá sám dle libosti (Instruktáž k použití dává zaměstnanec sportcentra). Potřebuje pouze vědět, kterou infrasaunu si zarezervoval a tím pádem kterou má využít. Na druhou stranu u masáží je princip opět stejný jako u předchozího typu služeb (spinning atd.). Pro zákazníka je nutností znát maséra, ke kterému se objedná a rovněž masérnu, do které si půjde masáž nechat vykonat.

3.2 Analýza cen a permanentek sportcentra

Cenu jednotlivých služeb si určuje majitel sportcentra a jsou pro všechny zákazníky jednotné. Ceny se ovšem mohou lišit v závislosti na konkrétních lekcích, což znamená, že například služba aerobik může mít lekci, která potrvá 60 minut, bude určena pro začátečníky a bude stát 90 Kč. Dále pak ta samá služba může mít 90 minutovou lekci, která bude pro pokročilé a jejíž cena bude 130 Kč. Mohou nastat také případy krátkodobé změny cen, například když se majitel rozhodne zvýšit ceny v období očekávaného zvýšeného zájmu o danou službu, čímž může být například státní svátek. Toto zvýšení ceny je ovšem pouze dočasné a po uplynutí vybraného termínu je cena vrácena do své původní podoby.

Veškeré, dříve popsané služby jsou hrazeny buďto v hotovosti na pobočce sportovního centra nebo prostřednictvím permanentních vstupenek, které mají různý charakter.

3.2.1 Vstupové permanentky

Tento typ permanentek je založený na zakoupení x vstupů na jeden typ služby za určitou sumu. Permanentku lze zakoupit například na spinning či H.E.A.T. Dále lze permanentky zakoupit na různé služby, které se cvičí v aerobním sále.

3.2.2 Kreditové permanentky

Kreditové permanentky jsou jednoduše založeny na zaplacení dané částky a následné využívání kreditu na libovolné služby. V závislosti na nabitých částkách jsou poskytovány procentuální slevy na všechny přístupné služby.

3.2.3 Časové permanentky

U tohoto typu permanentky má zákazník možnost po zaplacení dané sumy neomezeně navštěvovat danou službu po určité vymezenou dobu (např. platnost 3 měsíce). Časové vstupné se využívá výhradně pro návštěvu posilovny, která ovšem není součástí rezervačního systému. Časové permanentky budou ovšem do systému zahrnuty pro případ změny rozhodnutí majitele.

3.3 Shrnutí a strukturování funkčních požadavků na rezervační systém

Po důkladné diskuzi s provozovatelem sportcentra je možné identifikovat následující požadavky.

P1 – zákazník se registruje do systému prostřednictvím internetu se schopností ověření pravdivého a existujícího kontaktu.

P2 – zákazník se přihlásí/odhlásí do/ze systému, v případě zapomenutí hesla možnost jeho obnovení prostřednictvím e-mailu.

P3 – zákazník si může prohlížet dostupné informace na jeho vlastním účtu.

P4 – zákazník provádí v systému rezervace na různé služby.

P5 – zákazník si vybírá způsob zaplacení rezervované služby.

P6 – zákazník stornuje rezervaci povolenou dobu před uskutečněním služby.

P7 – zákazník si zobrazí seznam jeho rezervací.

P8 – zákazník si zobrazí seznam svých historických rezervací.

P9 – zákazník si upraví údaje na svém zákaznickém účtu (včetně změny hesla).

P10 – majitel může zavádět, odstraňovat a upravovat různé služby, které sportovní centrum poskytuje. Může si rovněž zobrazovat jejich detaily a informace, které jsou o těchto službách dostupné. Zobrazování informací může vykonat i zaměstnanec sportovního centra. Nemůže ovšem zavádět, odstraňovat a upravovat žádné služby.

P11 – majitel může v systému spravovat (vytvářet, upravovat a odstraňovat) instruktory, kteří vedou jednotlivé lekce. Je možné se rovněž podívat na dostupné informace o instruktorovi. Zobrazení informací může provádět i zaměstnanec.

P12 – majitel může přidávat, odstraňovat, upravovat místa výkonu (cvičiště), spolu s tím si je schopen zobrazit dostupné informace o cvičišti včetně veškerých hodin, které na daném cvičišti probíhají. Zobrazit si dostupné informace může i zaměstnanec.

P13 – majitel může spravovat různé typy permanentek, které jsou dostupné pro zákazníky v nabídce sportovního centra. Sportovní centrum poskytuje různé permanentky, jako například vstupová permanentka na spinning, která je platná 1 rok od zakoupení. Zaměstnanec centra může vidět seznam dostupných permanentek, nemůže ovšem provádět žádné administrativní úkony.

P14 – majitel nebo zaměstnanec přidává jednotky služeb (lekce) do harmonogramu tak, aby se na ně mohli zákazníci rezervovat. Může rovněž tyto jednotky upravovat a mazat či vytvářet dočasné změny.

P15 – majitel nebo zaměstnanec může vyhledávat v databázi uživatelů, rezervací nebo zakoupených permanentek.

P16 – majitel a zaměstnanec se přihlašují/odhlašují do/ze systému.

P17 – majitel může vytvářet nové účty s různými pravomocemi. Může vytvořit uživatele systému se stejnými pravomocemi, jako má on sám, nebo s pravomocemi řadového zaměstnance. Může vytvářet i obyčejné zákaznické účty.

P18 – zaměstnanec může vytvářet pouze účty na požádání zákazníků na pobočce sportovního centra. Jiné pravomoci než, mají zákazníci, udělovat nemůže.

P19 – majitel i zaměstnanec mohou na vyžádání zákazníka obnovit zákaznicko heslo tak, že je mu odesláno automaticky na e-mail.

P20 – majitel i zaměstnanec mohou zákazníka zarezervovat i zrušit rezervaci na požadovanou službu. V takovém případě ale nesmí cenu služeb odečíst z permanentky. Můžou si rovněž zobrazovat nastávající i již proběhlé rezervace zákazníka.

P21 – majitel nebo zaměstnanec mohou v detailu uživatelského účtu prohlížet údaje o zákaznících.

P22 – majitel i zaměstnanec mohou zákazníkovi vytvářet nové permanentky na různé služby. Dále mohou na již zavedené permanentky dobíjet kredit, či jej převádět na jiné permanentky.

P23 – majitel i zaměstnanec si mohou zobrazit jaké permanentky má aktuálně zákazník v provozu.

P24 – majitel a zaměstnanec si mohou zobrazit historii toho, co se dělo s kreditem na konkrétní zákaznické permanentce.

P25 – majitel i zaměstnanec mohou provádět veškeré zákaznické úkony, týkající se administrace vlastního účtu a vytváření rezervací s tím rozdílem že žádnou službu nejsou povinni hradit z permanentky.

4. Návrh rezervačního systému pro podnikatele

4.1 Modelování případů užití

4.1.1 Aktéři systému

Administrátor

Nejvýše postavený aktér systému. Má veškerá práva na administraci a zároveň je rozšířením aktéra „Správce pobočky“ (a rovněž tedy rozšířením aktéra „zákazník“). Provádí tedy rovněž všechny úkony prováděné zákazníkem (na svém uživatelském účtu).

Správce pobočky

Správce pobočky je rozšířením zákazníka s částečnými právy na administraci. Správce pobočky rovněž může provádět úkony, které provádí zákazník v rámci svého účtu.

Zákazník

Tento uživatel nemá žádná administrátorská práva, pouze spravuje částečně svůj vlastní účet a své rezervace.

Neregistrovaný zákazník

Aktér přicházející zvenčí. Registrací se z něj stává „zákazník“.

4.1.2 Případy užití

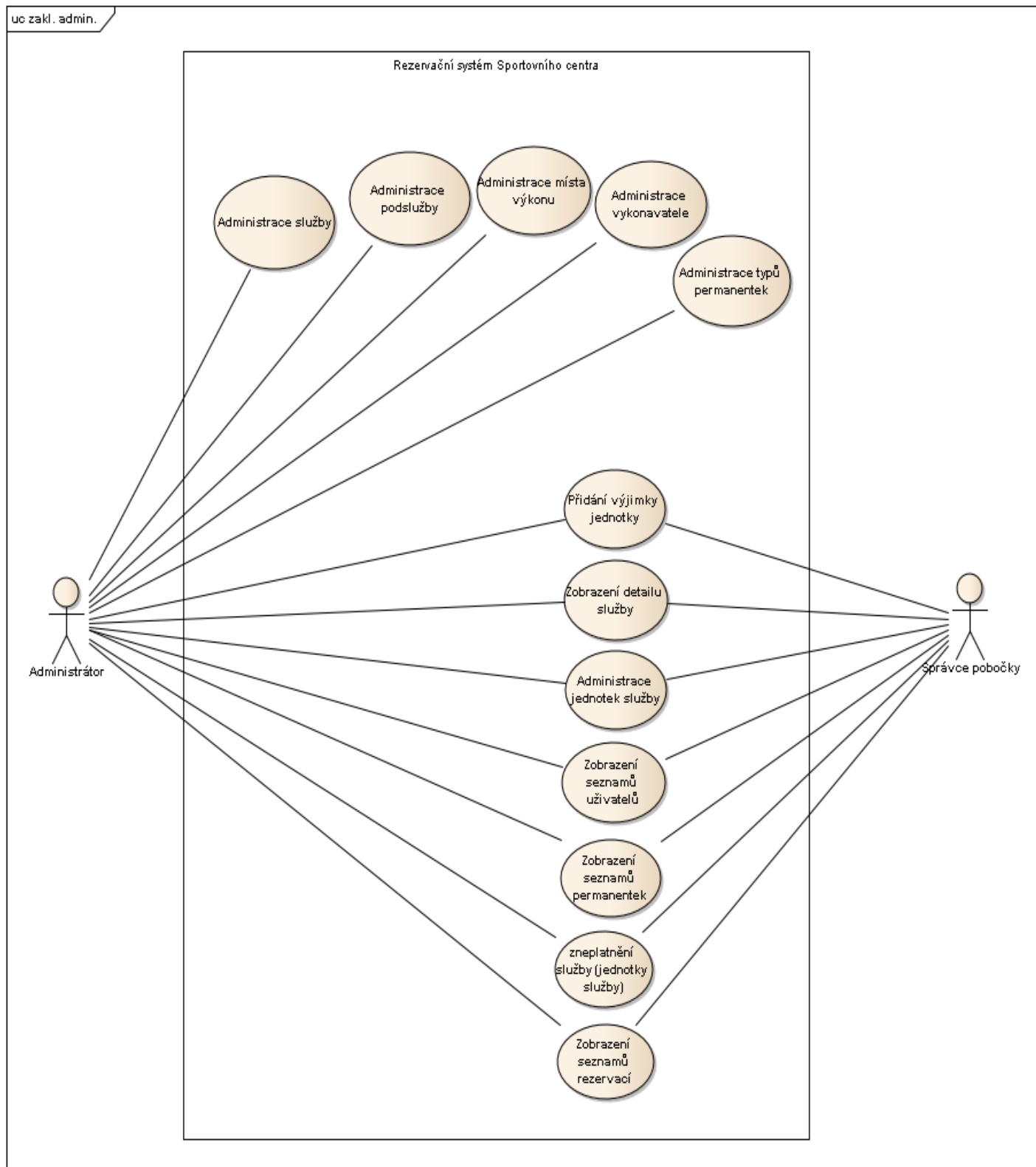
Podle předmětu zaměření byly případy užití sdruženy do tří skupin. První skupinou případů užití je **Základní administrace sportcentra**, do které byly zařazeny případy užití, týkající se administrace samotných služeb, nabízených sportcentrem. Jedná se například o přidávání a odebírání služeb, instruktorů, cvičišť apod. Další skupinou je **Administrace uživatelů**, která se týká manipulace se zákaznickými účty a vším, co s nimi souvisí, ovšem pouze prostřednictvím administrátorského rozhraní. Veškeré případy užití, které jsou zařazeny v těchto dvou skupinách, může spouštět pouze administrátor nebo správce pobočky sportcentra. Poslední skupinou je skupina **Zákaznické scénáře**. Tyto případy užití je schopen vykonávat jak zákazník, tak i administrátor a správce pobočky (samozřejmě v rámci svého zákaznického účtu). Patří zde případy užití, které se týkají přímého využívání rezervačního

systému z pohledu zákazníka, tzn. Registrování, přihlašování, vytváření a spravování vlastních rezervací apod.

Celkem bylo vytvořeno 34 případů užití. Detailní popis případů užití vyjadřují následující tabulky. Obsah tabulek byl sestaven dle vzoru popsaného v kapitole 2.2.2.1.

a) Základní administrace sportcentra

UC1 – Administrace služby:	Administrátor
UC2 – Administrace podslužby:	Administrátor
UC3 – Administrace místa výkonu:	Administrátor
UC4 – Administrace vykonavatelů:	Administrátor
UC5 – Administrace typů permanentek:	Administrátor
UC6 – Zneplatnění služby:	Administrátor, správce pobočky
UC7 – Zobrazení detailu služby:	Administrátor, správce pobočky
UC8 – Administrace jednotky služby:	Administrátor, správce pobočky
UC9 – Zobrazení seznamů uživatelů:	Administrátor, správce pobočky
UC10 – Zobrazení seznamů permanentek:	Administrátor, správce pobočky
UC11 – Zobrazení seznamů rezervací:	Administrátor, správce pobočky
UC12 – Přidání výjimky:	Administrátor, Správce pobočky



Obrázek 4.1: Diagram případů užití základní administrace

Případ užití: Administrace služby	
Id:	UC1
Stručný popis:	Zahrnuje vytvoření, úpravu a odstranění služby, poskytované sportcentrem.
Aktéři:	Administrátor
Výchozí podmínky:	<p>Uživatel musí mít roli administrátor. Musí být přihlášen do systému. <i>Speciální podmínky pro odstranění</i> Služba (resp. její jednotky) musí být před odstraněním zneplatněna. Služba nesmí mít přiřazené vstupové permanentky v hodnotě vyšší než je 0kč.</p>
Hlavní scénář:	<p><i>Vytvoření</i></p> <ol style="list-style-type: none"> 1. Administrátor si v administraci vybere záložku administrace služeb. 2. Klikne na možnost pro vytvoření nové služby. 3. Kliknutím na tlačítko uloží změny a vytvoří novou službu. 4. Aplikace vytvoří novou službu. <p><i>Úprava</i></p> <ol style="list-style-type: none"> 1. Administrátor si v administraci vybere službu, kterou chce upravit. 2. Kliknutím na tlačítko upravit zahájí změny. 3. změni potřebné údaje o službě. 4. Potvrdí změny kliknutím na tlačítko dokončení. 5. Aplikace provede potřebné změny. <p><i>Odstranění</i></p> <ol style="list-style-type: none"> 1. Administrátor si v administraci vybere službu, kterou chce odstranit. 2. Kliknutím na tlačítko odstranění služby zadá příkaz na smazání služby ze systému. 3. Aplikace odstraní službu ze systému.
Alternativní scénáře:	<p><i>Odstranění</i></p> <p>Alternativní scénář navazuje na krok č. 2 hlavního scénáře.</p> <ol style="list-style-type: none"> 1. Aplikace zjistí, že služba má stále aktivní permanentky. 2. Aplikace vypíše seznam aktivních permanentek do nového okna.
Výstupní podmínky:	V rezervačním systému je zavedena nová služba, případně je stávající služba odstraněna nebo upravena.
Poznámka:	<p>Pokud služba (její jednotky) není před odstraněním zneplatněna, tlačítko pro smazání služby není aktivní. Aplikace nepovolí odstranění služby, dokud nebudou administrátorem vyřešeny všechny aktivní permanentky.</p>

Tabulka 4.1: Popis případu užití administrace služby

Případ užití: Administrace podslužby	
Id:	UC2
Stručný popis:	Přidání nebo smazání či úpravu podslužby, vztahující se ke konkrétní službě sportcentra (např. částečná masáž).
Akteři:	Administrátor
Výchozí podmínky:	Uživatel musí mít roli administrátor. Musí být přihlášen do systému. <i>Podmínka Odstranění</i> Podslužba nesmí být aktuálně použita v rezervaci.
Hlavní scénář:	<p><i>Vytvoření</i></p> <ol style="list-style-type: none"> 1. Administrátor si v administraci vybere u konkrétní služby možnost zavést novou podslužbu. 2. Administrátor vyplní potřebné údaje o podslužbě. 3. Kliknutím na tlačítko uloží změny a vytvoří novou podslužbu. 4. Aplikace vytvoří novou podslužbu. <p><i>Úprava</i></p> <ol style="list-style-type: none"> 1. Administrátor si v administraci vybere upravit podslužbu v detailu služby. 2. Administrátor vyplní potřebné údaje o službě. 3. Kliknutím na tlačítko uloží změny. 4. Aplikace upraví podslužbu. <p><i>Odstranění</i></p> <ol style="list-style-type: none"> 1. Administrátor si v administraci vybere odstranit podslužbu v detailu služby. 2. Administrátor klikne na tlačítko pro odstranění podslužby. 3. Kliknutím na tlačítko pro dokončení potvrdí odstranění podslužby. 4. Aplikace smaže podslužbu ze systému.
Alternativní scénáře:	Žádné
Výstupní podmínky:	Z rezervačního systému je odstraněna podslužba.
Poznámka:	tzv. "podslužba" se týká pouze služby "masáže". Podslužbou může být například "masáž šíje" či "celková masáž". Podslužba se zavádí z důvodu rozlišení délky a ceny v rámci jedné služby. Zobrazit údaje o podslužbě lze kliknutím na ni v administraci.

Tabulka 4.2: Popis případu užití administrace poslužby

Případ užití: Administrace místa výkonu (cvičiště)	
Id:	UC3
Stručný popis:	Přidá, upraví parametry nebo odstraní místo výkonu služby. Místem výkonu se rozumí prostor určený k provozování dané aktivity, jako je například tělocvična, spinningový sál nebo maskérna.
Aktéři:	Administrátor
Výchozí podmínky:	Uživatel musí mít roli administrátor. Musí být přihlášen do systému. <i>Podmínka Odstranění</i> V místě výkonu nesmí být naplánovány žádné jednotky.
Hlavní scénář:	<p><i>Vytvoření</i></p> <ol style="list-style-type: none"> 1. Administrátor si v administraci míst výkonu služeb vybere možnost vytvořit nové místo výkonu. 2. Administrátor vyplní potřebné údaje. 3. Kliknutím na tlačítko uloží změny. 4. Aplikace vytvoří nové místo výkonu. <p><i>Úprava</i></p> <ol style="list-style-type: none"> 1. Administrátor si v administraci míst výkonu služeb vybere upravit místo výkonu v detailu služby. 2. Administrátor vyplní potřebné údaje. 3. Kliknutím na tlačítko uloží změny. 4. Aplikace upraví parametry místa výkonu. <p><i>Odstranění</i></p> <ol style="list-style-type: none"> 1. Administrátor si v administraci míst výkonu služeb vybere odstranit místo výkonu. 2. Administrátor klikne na tlačítko pro odstranění místa výkonu. 3. Kliknutím na tlačítko pro dokončení. 4. Aplikace smaže místo výkonu služby.
Alternativní scénáře:	Žádné
Výstupní podmínky:	Z rezervačního systému je odstraněno místo výkonu.
Poznámka:	Kliknutím na místo výkonu v jejich seznamu se zobrazí detailní rozpis služeb v harmonogramu daného místa výkonu.

Tabulka 4.3: Popis případu užití administrace místa výkonu

Případ užití: Administrace vykonavatelů služby (instruktorů)	
Id:	UC4
Stručný popis:	Přidává, upravuje a maže vykonavatele (trenéry, cvičitele, maséry). Tito vykonavatelé jsou obecně vedeni v administraci sportcentra, jelikož jeden instruktor může trénovat více služeb. Například cvičitel pilates může zároveň trénovat i aerobik apod.
Aktéři:	Administrátor
Výchozí podmínky:	Uživatel musí mít roli administrátor. Musí být přihlášen do systému. <i>Podmínka pro Odstranění</i> Daný vykonavatel nesmí mít aktivní jednotky.
Hlavní scénář:	<p><i>Vytvoření</i></p> <ol style="list-style-type: none"> 1. Administrátor si v administraci vykonavatelů (instruktorů) vybere možnost zavést nového vykonavatele. 2. Administrátor vyplní potřebné údaje. 3. Kliknutím na tlačítko uloží změny. 4. Aplikace vytvoří nového vykonavatele služby. <p><i>Úprava</i></p> <ol style="list-style-type: none"> 1. Administrátor si v administraci vykonavatelů (instruktorů) vybere upravit vykonavatele v detailu služby. 2. Administrátor vyplní potřebné údaje. 3. Kliknutím na tlačítko uloží změny. 4. Aplikace upraví parametry vykonavatele. <p><i>Odstranění</i></p> <ol style="list-style-type: none"> 1. Administrátor si v administraci vykonavatelů (instruktorů) vybere odstranit vykonavatele. 2. Administrátor klikne na tlačítko pro odstranění. 3. Kliknutím na tlačítko pro dokončení. 4. Aplikace smaže vykonavatele ze seznamu.
Alternativní scénáře:	Žádné
Výstupní podmínky:	Z rezervačního systému je odstraněn vykonavatel (instruktor).
Poznámka:	Kliknutím na název vykonavatele v jejich seznamu se zobrazí jeho detaily, resp. k jakým jednotkám a službám jsou přiřazeni.

Tabulka 4.4: Popis případu užití administrace vykonavatelů služby

Případ užití: Administrace typů permanentek	
Id:	UC5
Stručný popis:	Vytvoří, edituje nebo odstraní nový typ permanentky, který pak bude možno přidávat uživatelům. Příkladem je vstupová permanentky na pilates s platností 1 rok.
Aktéři:	Administrátor
Výchozí podmínky:	Uživatel musí mít roli administrátor. Musí být přihlášen do systému. <i>Odstranění</i> Permanentka nesmí být přiřazena k žádnému zákazníkovi.
Hlavní scénář:	<i>Vytvoření</i> 1. Uživatel si v administraci zadá možnost vytvořit nový typ permanentky. 2. Vyplní údaje. 3. Klikne na tlačítko pro uložení permanentky. 4. Aplikace uloží permanentku do seznamu typů permanentek. <i>Úprava</i> 1. Uživatel si v administraci permanentek klikne na tlačítko upravit u vybraného typu permanentky. 2. Zadá úpravy. 3. Aplikace provede změny. <i>Odstranění</i> 1. Administrátor klikne v seznamu typů permanentek na tlačítko pro odstranění konkrétního typu permanentky. 2. Potvrdí, že chce permanentku skutečně odstranit. 3. Aplikace odstraní permanentku ze seznamu.
Alternativní scénáře:	<i>Odstranění</i> (navazuje na krok 1. hlavního scénáře) 1. Aplikace najde tento typ permanentky navázán na uživatelské účty. 2. Aplikace vypíše seznam zákaznických účtů.
Výstupní podmínky:	Uživateli byl zobrazen požadovaný seznam.
Poznámka:	<i>Poznámka k alternativnímu scénáři odstranění</i> Pokud je na zákaznickém účtu u daného typu permanentky zůstatek 0 Kč, systém automaticky smaže permanentku a odešle uživateli notifikaci.

Tabulka 4.5: Popis případu užití administrace typů permanentek

Případ užití: Zneplatnění služby (jednotky služby)	
Id:	UC6
Stručný popis:	V případě, že sportcentrum ruší jednotky služeb ze svého sortimentu, zneplatní službu od určitého zadaného data, čímž zruší možnost je rezervovat. Jednotky lze zneplatňovat stejně jako v případě přidávání výjimek – hromadně podle společného znaku jednotky (například instruktora nebo cvičiště).
Aktéři:	Administrátor
Výchozí podmínky:	Uživatel musí mít roli administrátor. Musí být přihlášen do systému.
Hlavní scénář:	<ol style="list-style-type: none"> 1. Administrátor si v administraci vybere možnost zneplatnění konkrétní jednotky služby. 2. Zadá údaje pro zneplatnění jednotky služby (datum). 3. Klikne na tlačítko dokončení zneplatnění. 4. Aplikace zneplatní službu a rozešle všem zákazníkům oznámení o zneplatnění. Aktivní rezervace na období po datu zneplatnění budou zrušeny a zákazníkům bude odeslán oznamovací e-mail.
Alternativní scénáře:	Žádné
Výstupní podmínky:	Daná služba nebo její část bude zneplatněna k zadanému datu.
Poznámka:	Aplikace bude odesílat informace o zneplatnění služby (jednotky služby). Zákazníci budou vyzváni k osobnímu vyřešení situace ohledně zbývajících kreditů na permanentkách, vztahujících se ke konkrétní službě, pokud se jedná o zneplatnění všech jednotek služby.

Tabulka 4.6: Popis případu užití zneplatnění služby

Případ užití: Zobrazení detailu služby	
Id:	UC7
Stručný popis:	Zobrazí detail požadované služby (jaké obsahuje podslužby, jednotky atd.)
Aktéři:	Administrátor, správce pobočky
Výchozí podmínky:	Uživatel musí mít roli administrátor. Musí být přihlášen do systému.
Hlavní scénář:	<ol style="list-style-type: none"> 1. Administrátor si v administraci vybere administraci služeb. 2. Vybere ze seznamu službu, kterou chce zobrazit. 3. Aplikace zobrazí detail služby.
Alternativní scénáře:	Žádné
Výstupní podmínky:	Zobrazení detailu služby.
Poznámka:	

Tabulka 4.7: Popis případu užití zobrazení detailu služby

Případ užití: Administrace jednotky služby	
Id:	UC8
Stručný popis:	Vytvoří, upraví nebo vymaže jednotku konkrétní služby, čímž se myslí například nová hodina spinningu či nové cvičení aerobiku.
Aktéři:	Administrátor, správce pobočky
Výchozí podmínky:	Uživatel musí mít roli administrátor nebo správce pobočky. Uživatel musí být přihlášen do systému.
Hlavní scénář:	<p><i>Vytvoření</i></p> <ol style="list-style-type: none"> 1. Uživatel si v administraci služeb zobrazí detail služby, které chce přiřadit novou jednotku. 2. Klikne na tlačítko pro přidání nové jednotky. 3. Zadá potřebné atributy (např. časové údaje o jednotce). 4. Klikne na tlačítko pro dokončení. 5. Aplikace vytvoří novou jednotku služby. <p><i>Úprava</i></p> <ol style="list-style-type: none"> 1. Uživatel si v administraci vyhledá jednotku služby, kterou chce upravit. 2. Klikne na tlačítko pro upravení dané jednotky. 3. Upraví atributy. 4. Klikne na tlačítko pro dokončení. 5. Aplikace upraví vybranou jednotku služby. Pokud jsou na tuto jednotku již nějaké aktivní rezervace, zákazníkům bude odeslán informativní e-mail se změnami. <p><i>Odstranění</i></p> <ol style="list-style-type: none"> 1. Uživatel si vybere jednotku služby, kterou plánuje odstranit. 2. Klikne na tlačítko pro odstranění jednotky. 3. Aplikace odstraní vybranou jednotku. Zároveň budou zrušeny všechny aktivní rezervace a zákazníci budou o změnách informováni e-mailem.
Alternativní scénáře:	
Výstupní podmínky:	V harmonogramu bude k dispozici nová jednotka služby.
Poznámka:	

Tabulka 4.8: Popis případu užití administrace jednotky služby

Případ užití: Zobrazení seznamů uživatelů	
Id:	UC9
Stručný popis:	Zobrazí seznam uživatelů podle vybraných kritérií.
Aktéři:	Administrátor, správce pobočky
Výchozí podmínky:	Uživatel musí mít roli administrátor nebo správce pobočky. Musí být přihlášen do systému.
Hlavní scénář:	<ol style="list-style-type: none"> 1. Uživatel si v administraci vybere záložku uživatelé. 2. Vybere požadovaná kritéria, podle nichž chce seznam filtrovat. 3. Klikne na tlačítko pro zobrazení seznamu. 4. Aplikace zobrazí požadované údaje.
Alternativní scénáře:	žádné
Výstupní podmínky:	Uživateli byl zobrazen požadovaný seznam.
Poznámka:	

Tabulka 4.9: Popis případu užití zobrazení seznamu uživatelů

Případ užití: Zobrazení seznamů permanentek	
Id:	UC10
Stručný popis:	Zobrazí seznam permanentek podle vybraných kritérií.
Aktéři:	Administrátor, správce pobočky
Výchozí podmínky:	Uživatel musí mít roli administrátor nebo správce pobočky. Musí být přihlášen do systému.
Hlavní scénář:	<ol style="list-style-type: none"> 1. Uživatel si v administraci zadá možnost zobrazit seznamy permanentek. 2. Vybere požadovaná kritéria, podle nichž chce seznamy filtrovat. 3. Klikne na tlačítko pro zobrazení seznamu 4. Aplikace zobrazí požadované údaje.
Alternativní scénáře:	žádné
Výstupní podmínky:	Uživateli byl zobrazen požadovaný seznam.
Poznámka:	

Tabulka 4.10: Popis případu užití zobrazení seznamu permanentek

Případ užití: Zobrazení seznamů rezervací	
Id:	UC11
Stručný popis:	Zobrazí seznam permanentek podle vybraných kritérií.
Akteři:	Administrátor, správce pobočky
Výchozí podmínky:	Uživatel musí mít roli administrátor nebo správce pobočky. Musí být přihlášen do systému.
Hlavní scénář:	1. Uživatel si v administraci zadá možnost zobrazit seznamy permanentek. 2. Vybere požadovaná kritéria, podle nichž chce seznamy filtrovat. 3. Klikne na tlačítko pro zobrazení seznamu 4. Aplikace zobrazí požadované údaje.
Alternativní scénáře:	Žádné
Výstupní podmínky:	Uživateli byl zobrazen požadovaný seznam.
Poznámka:	

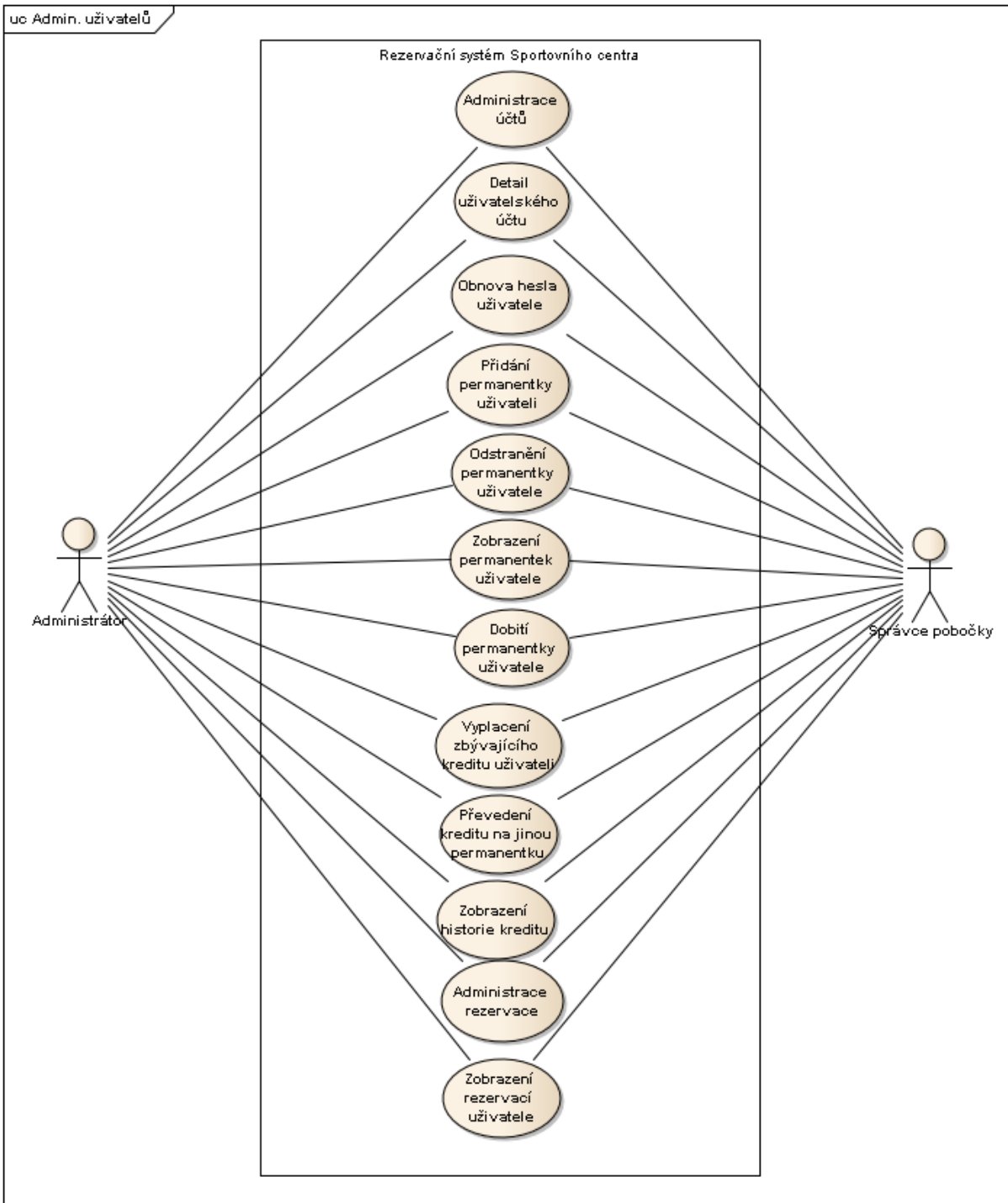
Tabulka 4.11: Popis případu užití zobrazení seznamu rezervací

Případ užití: Přidání výjimky	
Id:	UC12
Stručný popis:	Vytvoří výjimku v dostupnosti jednotky služby (například konkrétní hodiny spinningu). Tato výjimka řeší dočasné změny v attributech jednotky.
Akteři:	Administrátor, Správce pobočky
Výchozí podmínky:	Uživatel musí mít roli administrátor. Musí být přihlášen do systému.
Hlavní scénář:	1. Administrátor si v administraci vybere záložku časové výjimky. 2. Vyplní údaje o tom, k čemu se bude výjimka vztahovat. 3. Potvrdí tvorbu časové výjimky. 4. Aplikace vytvoří časovou výjimku a znemožní rezervace. Zároveň budou zrušeny aktivní rezervace a zákazníkům budou rozeslány informativní e-maily.
Alternativní scénáře:	Žádné
Výstupní podmínky:	Rezervace na požadovanou dobu, jednotku či sport budou znemožněny a stávající rezervace budou zrušeny.
Poznámka:	

Tabulka 4.12: Popis případu užití přidání výjimky

b) Administrace uživatelů

UC13 – Administrace účtů:	Administrátor, správce pobočky
UC14 – Detail uživatelského účtu:	Administrátor, správce pobočky
UC15 – Obnova hesla uživatele:	Administrátor, správce pobočky
UC16 – Přidání permanentky uživateli:	Administrátor, správce pobočky
UC17 – Odstranění permanentky uživatele:	Administrátor, správce pobočky
UC18 – Zobrazení permanentek uživatele:	Administrátor, správce pobočky
UC19 – Dobití permanentky uživatele:	Administrátor, správce pobočky
UC20 – Vyplacení zbývajcího kreditu uživateli:	Administrátor, správce pobočky
UC21 – Převedení kreditu na jinou permanentku:	Administrátor, správce pobočky
UC22 – Zobrazení historie kreditu:	Administrátor, správce pobočky
UC23 – Administrace rezervace:	Administrátor, správce pobočky
UC24 – Zobrazení rezervací uživatele:	Administrátor, správce pobočky



Obrázek 4.2: Diagram případů užití administrace uživatelů

Případ užití: Administrace účtů	
Id:	UC13
Stručný popis:	Založí, upraví nebo odstraní se libovolný uživatelský účet přes administraci (pouze administrátorem či správcem pobočky).
Aktéři:	Administrátor, správce pobočky
Výchozí podmínky:	Uživatel musí mít roli administrátor nebo správce pobočky. Musí být přihlášen do systému. <i>Speciálně pro odstranění</i> Uživatel nesmí mít aktivní rezervace Uživatel nesmí mít zůstatky na permanentkách.
Hlavní scénář:	<p><i>Založení</i></p> <ol style="list-style-type: none"> 1. Uživatel zadá v administraci uživatelů možnost vytvořit nového uživatele. 2. Vyplní potřebné údaje. 3. Klikne na tlačítko pro dokončení. 4. Aplikace Vytvoří nový uživatelský účet. <p><i>Úprava</i></p> <ol style="list-style-type: none"> 1. Uživatel si v administraci vyhledá uživatele, jemuž chce upravit účet. 2. Vybere možnost upravit údaje v záložce osobních údajů. 3. Upraví potřebné údaje na účtu. 4. Klikne na tlačítko pro uložení. 5. Aplikace provede změny v údajích a notifikuje vlastníka uživatelského účtu e-mailem. <p><i>Odstranění</i></p> <ol style="list-style-type: none"> 1. Uživatel si v administraci vyhledá uživatele, kterého chce vymazat ze systému. 2. Klikne na tlačítko pro odstranění uživatele. 3. Aplikace odstraní uživatele ze systému.
Alternativní scénáře:	žádné
Výstupní podmínky:	Byl vytvořen nový uživatelský účet.
Poznámka:	Administrátor má možnost udělovat všechny role (rolí administrátor, správce pobočky a zákazník). Správce pobočky má povoleno udělovat pouze role zákazník. To platí rovněž o další manipulaci s účty (úprava a odstranění).

Tabulka 4.13: Popis případu užití administrace účtů

Případ užití: Detail uživatelského účtu	
Id:	UC14
Stručný popis:	Zobrazí náhled uživatelského účtu.
Aktéři:	Administrátor, správce pobočky
Výchozí podmínky:	Uživatel musí mít roli administrátor nebo správce pobočky. Musí být přihlášen do systému.
Hlavní scénář:	1. Uživatel si v administraci vyhledá uživatele, kterého chce zobrazit 2. Klikne na tlačítko pro zobrazení detailu. 3. Aplikace zobrazí uživatelský účet.
Alternativní scénáře:	Žádné
Výstupní podmínky:	Byl zobrazen detail uživatelského účtu.
Poznámka:	

Tabulka 4.14: Popis případu užití detail uživatelského účtu

Případ užití: Obnova hesla uživatele	
Id:	UC15
Stručný popis:	Obnoví heslo vybraného uživatelského účtu.
Aktéři:	Administrátor, správce pobočky
Výchozí podmínky:	Uživatel musí mít roli administrátor nebo správce pobočky. Musí být přihlášen do systému.
Hlavní scénář:	1. Uživatel si v administraci vyhledá uživatele, kterému chce obnovit heslo a zobrazí si jeho detail. 2. Vybere možnost obnovení hesla uživateli. 3. Aplikace obnoví heslo vybranému uživatelskému účtu a nové heslo bude zasláno uživateli na jeho registrační e-mail.
Alternativní scénáře:	žádné
Výstupní podmínky:	Heslo uživatelského účtu bylo změněno.
Poznámka:	

Tabulka 4.15: Popis případu užití obnova hesla uživatele

Případ užití: Přidání permanentky uživateli	
Id:	UC16
Stručný popis:	Vytvoří novou permanentku uživateli.
Aktéři:	Administrátor, správce pobočky
Výchozí podmínky:	Uživatel musí mít roli administrátor nebo správce pobočky. Uživatel musí být přihlášen do systému.
Hlavní scénář:	<ol style="list-style-type: none"> 1. Uživatel si v administraci vyhledá zákazníka, kterému chce permanentku vytvořit. 2. Klikne na tlačítko pro vytvoření permanentky. 3. Zadá potřebné údaje o permanentce. 4. Potvrdí tlačítkem pro dokončení. 5. Aplikace vytvoří zákazníkovi novou permanentku.
Alternativní scénáře:	<p>Navazuje na krok č. 2 hlavního scénáře.</p> <ol style="list-style-type: none"> 1. Aplikace zjistí, že již existuje stejný typ permanentky na účtu vybraného zákazníka. 2. Uživatel bude o situaci obeznámen (např. dialogové okno) a bude vyzván k úpravě stávající permanentky.
Výstupní podmínky:	Uživateli byla přidána nová permanentka dle jeho požadavků.
Poznámka:	Uživateli lze v detailu navíc upravit datum platnosti permanentky.

Tabulka 4.16: Popis případu užití přidání permanentky uživateli

Případ užití: Odstranění permanentky uživatele	
Id:	UC17
Stručný popis:	Odstraní permanentku z účtu zákazníka.
Aktéři:	Administrátor, správce pobočky
Výchozí podmínky:	Uživatel musí mít roli administrátor nebo správce pobočky. Musí být přihlášen do systému. Na permanentce nesmí být žádný zbývajících kredit.
Hlavní scénář:	<ol style="list-style-type: none"> 1. Uživatel si v administraci vyhledá zákazníka, jemuž chce odstranit permanentku. 2. Vybere permanentku, kterou chce odstranit. 3. Klikne na tlačítko pro odstranění permanentky. 4. Aplikace odstraní zákazníkovi permanentku z účtu a odešle notifikaci na e-mail.
Alternativní scénáře:	Žádné
Výstupní podmínky:	Permanentka byla odstraněna ze zákaznického účtu.
Poznámka:	Pokud bude na permanentce zůstatek větší než je 0 Kč, tlačítko pro odstranění permanentky nebude aktivní.

Tabulka 4.17: Popis případu užití odstranění permanentky uživatele

Případ užití: Zobrazení permanentek uživatele	
Id:	UC18
Stručný popis:	Zobrazí permanentky uživatele.
Aktéři:	Administrátor, správce pobočky
Výchozí podmínky:	Uživatel musí mít roli administrátor nebo správce pobočky. Musí být přihlášen do systému.
Hlavní scénář:	1. Uživatel si v administraci vybere zákazníka, jehož permanentky chce zobrazit. 2. Klikne na tlačítko pro zobrazení permanentek. 3. Aplikace zobrazí permanentky uživatele.
Alternativní scénáře:	Žádné
Výstupní podmínky:	Byly zobrazeny permanentky uživatele.
Poznámka:	

Tabulka 4.18: Popis případu užití zobrazení permanentek uživatele

Případ užití: dobítí permanentky uživatele	
Id:	UC19
Stručný popis:	Dobije na existující permanentku nový kredit.
Aktéři:	Administrátor, správce pobočky
Výchozí podmínky:	Uživatel musí mít roli administrátor nebo správce pobočky. Musí být přihlášen do systému.
Hlavní scénář:	1. Uživatel si v administraci vybere zákazníka, jemuž chce dobít kredit. 2. Vybere permanentku na zákaznickém účtu, na kterou chce přidat kredit a stiskne tlačítko pro přidání kreditu. 3. Vyplní údaje. 4. Uloží změny. 5. Aplikace provede změny v hodnotě permanentky a notifikuje uživatele e-mailem.
Alternativní scénáře:	žádné
Výstupní podmínky:	Permanentka byla upravena.
Poznámka:	

Tabulka 4.19: Popis případu užití dobítí permanentky uživatele

Případ užití: Vyplacení zbývajícího kreditu uživateli	
Id:	UC20
Stručný popis:	Provede úkon s popisem vyplacení zbývajícího kreditu z permanentky a změni hodnotu permanentky.
Aktéři:	Administrátor, správce pobočky
Výchozí podmínky:	Uživatel musí mít roli administrátor nebo správce pobočky. Musí být přihlášen do systému.
Hlavní scénář:	<ol style="list-style-type: none"> 1. Uživatel si v administraci vybere zákazníka, jemuž chce vyplatit zbylý kredit. 2. Vybere typ permanentky na zákaznickém účtu, kterou chce vyplatit. 3. Klikne na tlačítko vyplatit kredit. 4. Do zadávacího pole vyplní, do jaké výše chce kredit uživateli vyplatit. 5. Potvrzovacím tlačítkem dokončí akci. 6. Aplikace provede úkon a odečte veškerý kredit z permanentky.
Alternativní scénáře:	Žádné
Výstupní podmínky:	Požadovaný kredit byl odstraněn z permanentky.
Poznámka:	

Tabulka 4.20: Popis případu užití vyplacení zbývajícího kreditu uživateli

Případ užití: Převedení kreditu na jinou permanentku	
Id:	UC21
Stručný popis:	Provede úkon s popisem převedení zbylého kreditu na jinou permanentku.
Aktéři:	Administrátor, správce pobočky
Výchozí podmínky:	Uživatel musí mít roli administrátor nebo správce pobočky. Musí být přihlášen do systému.
Hlavní scénář:	<ol style="list-style-type: none"> 1. Uživatel si v administraci vybere zákazníka, jemuž chce převést kredit. 2. Vybere typ permanentky na zákaznickém účtu, ze které chce převést kredit. 3. Klikne na tlačítko pro převedení kreditu. 4. Do zadávacího pole vyplní výši kreditu, který bude z permanentky převeden. Dále vybere ze seznamu permanentek tu, na kterou bude kredit převeden a do posledního pole vyplní částku, která bude připsána na vybranou permanentku. 5. Klikne na tlačítko pro dokončení akce. 6. Aplikace provede změny a notifikuje uživatele na e-mail.
Alternativní scénáře:	Žádné
Výstupní podmínky:	Z permanentky byl odepsán kredit a na druhou vybranou permanentku byl připsán.

Poznámka:	Kredit se odepisuje i připisuje na jednotlivých permanentkách v jednotkách, v jakých je veden. Převod mezi výší kreditu jednotlivých permanentek je věc provozovatele. (Jinými slovy jakou částku provozovatel přičte na kreditovou permanentku např. za 5 vstupů odečtených ze vstupové permanentky na spinning je záležitost domluvy mezi zaměstnanci sportcentra).
-----------	---

Tabulka 4.21: Popis případu užití převedení kreditu na jinou permanentku

Případ užití: Zobrazení historie kreditu	
Id:	UC22
Stručný popis:	Zobrazí historii kreditu vybrané permanentky.
Akteři:	Administrátor, správce pobočky
Výchozí podmínky:	Uživatel musí mít roli administrátor nebo správce pobočky. Musí být přihlášen do systému.
Hlavní scénář:	<ol style="list-style-type: none"> 1. Uživatel si v administraci vyhledá zákazníka, jehož kredit chce zobrazit. 2. Zobrazí si permanentky uživatele a klikne na možnost zobrazení historie u vybrané permanentky. 3. Aplikace zobrazí historii permanentky.
Alternativní scénáře:	Žádné
Výstupní podmínky:	Historie permanentky byla zobrazena.
Poznámka:	

Tabulka 4.22: Popis případu užití zobrazení historie kreditu

Případ užití: Administrace rezervace	
Id:	UC23
Stručný popis:	Vytvoří nebo zruší rezervaci vybranému uživateli.
Akteři:	Administrátor, správce pobočky
Výchozí podmínky:	Uživatel musí mít roli administrátor nebo správce pobočky. Musí být přihlášen do systému.
Hlavní scénář:	<p><i>Vytvoření</i></p> <ol style="list-style-type: none"> 1. Uživatel si v administraci vybere detail zákazníka, kterému chce vytvořit rezervaci. 2. Klikne na tlačítko pro vytvoření nové rezervace v záložce pro administraci rezervací zákazníka. 3. Vybere případnou podslužbu a klikne na tlačítko pro potvrzení rezervace. 4. Aplikace vytvoří novou rezervaci uživateli. <p><i>Zrušení</i></p> <ol style="list-style-type: none"> 1. Uživatel si v administraci vybere zákazníka, jemuž chce zrušit rezervaci. 2. Zobrazí si rezervace uživatele.

	3. U vybrané rezervace klikne na tlačítko pro její zrušení. 4. Aplikace odstraní rezervaci a odešle notifikaci na e-mail. Zároveň navrátí odečtený kredit uživateli na permanentku, ze které byl kredit při rezervaci stržen.
Alternativní scénáře:	Žádné
Výstupní podmínky:	Vybraný uživatel má vytvořenou nebo zrušenou rezervaci.
Poznámka:	Pokud rezervaci vytváří administrátor nebo správce pobočky, nesmí být cena služby odečtena zákazníkovi z permanentky. Pokud byla takto vytvořená rezervace zároveň zrušena, nebude kredit samozřejmě ani přičten na žádnou permanentku.

Tabulka 4.23: Popis případu užití administrace rezervace

Případ užití: Zobrazení rezervací uživatele	
Id:	UC24
Stručný popis:	Zobrazí aktivní rezervace uživatele.
Aktéři:	Administrátor, správce pobočky
Výchozí podmínky:	Uživatel musí mít roli administrátor nebo správce pobočky. Musí být přihlášen do systému.
Hlavní scénář:	1. Uživatel si v administraci vybere zákazníka, jehož rezervace chce zobrazit. 2. Klikne na tlačítko pro zobrazení rezervací. 3. Aplikace zobrazí rezervace uživatele.
Alternativní scénáře:	Žádné
Výstupní podmínky:	Byly zobrazeny rezervace uživatele.
Poznámka:	

Tabulka 4.24: Popis případu užití zobrazení rezervací uživatele

c) Zákaznické scénáře

UC25 – Registrace zákaznického účtu: neregistrovaný zákazník

UC26 – Úprava zákaznického účtu: zákazník

UC27 – Změna hesla: zákazník

UC28 – Obnova hesla: zákazník

UC29 – Přihlášení zákazníka do systému: zákazník

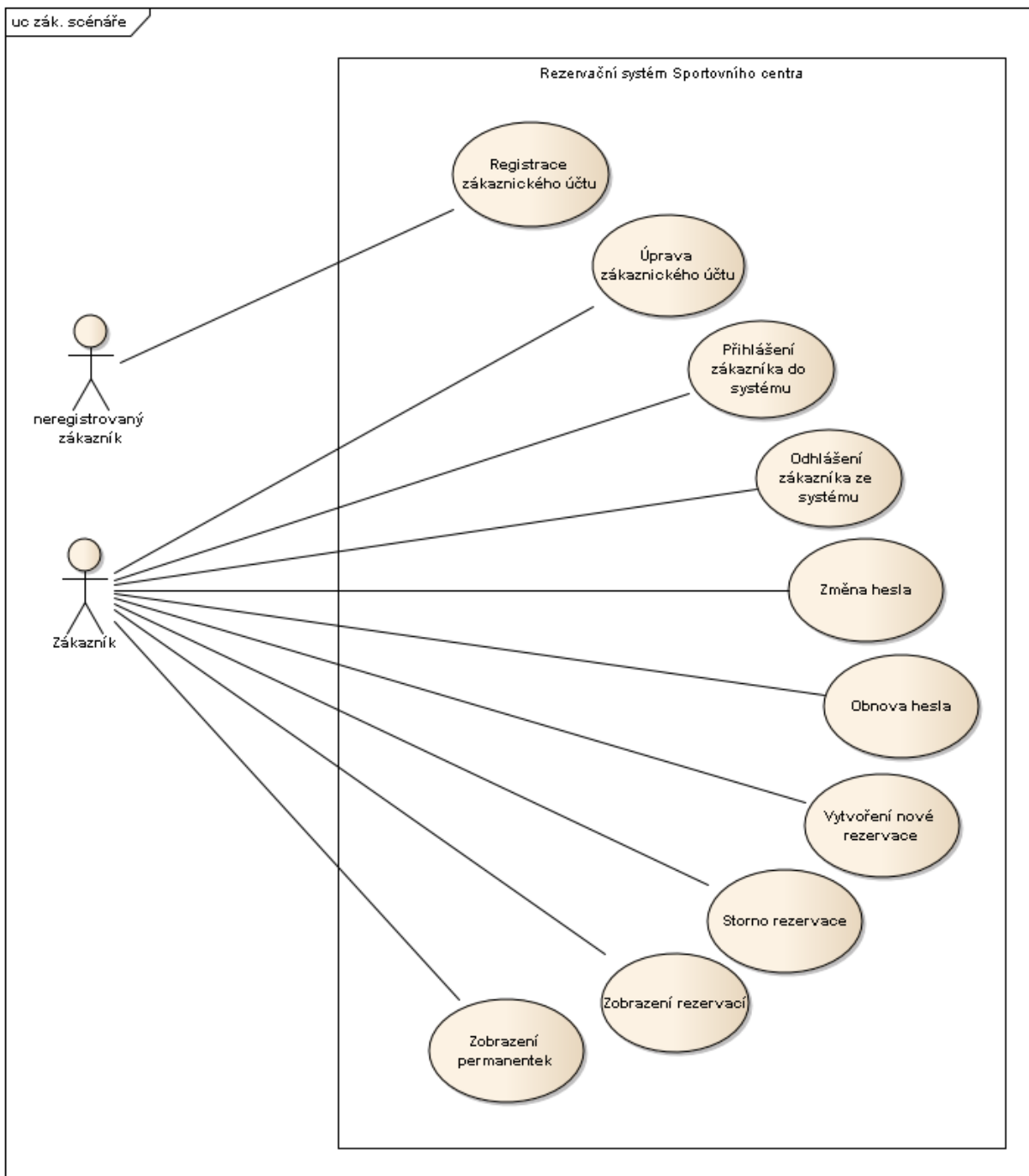
UC30 – Odhlášení zákazníka ze systému: zákazník

UC31 – Vytvoření rezervace: zákazník

UC32 – Storno rezervace: zákazník

UC33 – Zobrazení rezervací: zákazník

UC34 – Zobrazení permanentek: zákazník



Obrázek 4.3: Diagram případů užití zákaznických scénářů

Případ užití: Registrace zákaznického účtu	
Id:	UC25
Stručný popis:	Zaregistruje do systému neregistrovaného zákazníka.
Aktéři:	neregistrovaný zákazník
Výchozí podmínky:	Uživatel musí mít roli neregistrovaný zákazník a při registraci musí vyplnit veškeré povinné údaje.
Hlavní scénář:	<ol style="list-style-type: none"> 1. Neregistrovaný zákazník si v úvodním okně rezervačního systému vybere možnost se zaregistrovat. 2. Vyplní potřebné povinné, případně volitelné údaje. 3. Klikne na tlačítko pro dokončení registrace. 4. Aplikace vytvoří účet a odešle potvrzovací e-mail s odkazem. 5. Zákazník klikne na odkaz v e-mailu a dokončí registraci. 6. Aplikace dokončí registraci a zpřístupní zákaznický účet.
Alternativní scénáře:	<p><i>Alternativní scénář 1 navazuje na krok 4 hl. scénáře</i></p> <ol style="list-style-type: none"> 1. zákazník neaktivoval potvrzovací odkaz v požadované době - registrace byla zrušena. <p><i>Alternativní scénář 2 navazuje na krok 4 hl. scénáře</i></p> <ol style="list-style-type: none"> 1. Pokud zákazník nepotvrdil registraci a zároveň nevypršela doba pro dokončení registrace, zákazník se na úvodní stránce přihlásí svými přihlašovacími údaji. 2. Poté mu bude zobrazena stránka s oznámením o nedokončení registrace a s tlačítkem pro opětovné zaslání potvrzovacího e-mailu. 3. Zákazník klikne na tlačítko pro zaslání potvrzovacího e-mailu. 4. Aplikace znovu odešle potvrzovací e-mail. 5. Zákazník v e-mailu potvrdí registraci. 6. Aplikace dokončí registraci uživatele.
Výstupní podmínky:	Byl vytvořen nový zákaznický účet.
Poznámka:	

Tabulka 4.25: Popis případu užití registrace zákaznického účtu

Případ užití: Úprava zákaznického účtu	
Id:	UC26
Stručný popis:	Úprava údajů na vlastním uživatelském účtu.
Aktéři:	Zákazník
Výchozí podmínky:	<p>Uživatel musí mít roli zákazník.</p> <p>Uživatel musí být přihlášen do systému.</p>
Hlavní scénář:	<ol style="list-style-type: none"> 1. Zákazník si v prostředí svého účtu vybere možnost upravit v záložce „můj účet“. 2. Provede potřebné úpravy. 3. Klikne na tlačítko pro uložení změn. 4. Aplikace uloží změny údajů zákaznického účtu.
Alternativní scénáře:	žádné
Výstupní podmínky:	Údaje zákazníka byly upraveny.
Poznámka:	

Tabulka 4.26: Popis případu užití úprava zákaznického účtu

Případ užití: Změna hesla	
Id:	UC27
Stručný popis:	Zákazník si změní heslo na svém účtu.
Aktéři:	Zákazník
Výchozí podmínky:	Uživatel musí mít roli zákazník. Uživatel musí být přihlášen do systému.
Hlavní scénář:	1. Zákazník si v prostředí svého účtu vybere možnost změnit heslo. 2. Vyplní požadované údaje. 3. Klikne na tlačítko pro uložení změn. 4. Aplikace uloží nové heslo k zákaznickému účtu.
Alternativní scénáře:	Navazuje na krok 3 hlavního scénáře 1. Požadované údaje jsou zákazníkem vyplněny nesprávně. 2. Aplikace vyhodí chybovou hlášku a vyzve zákazníka k opětovnému zadání údajů.
Výstupní podmínky:	Heslo k zákaznickému účtu bylo změněno.
Poznámka:	Zadání nesprávných údajů = např. nové heslo je příliš krátké, staré heslo je zadáno nesprávně.

Tabulka 4.27: Popis případu užití změna hesla

Případ užití: Obnova hesla	
Id:	UC28
Stručný popis:	V případě zapomenutí hesla jej obnoví a nové heslo zašle na e-mail.
Aktéři:	Zákazník
Výchozí podmínky:	Uživatel musí mít roli zákazník.
Hlavní scénář:	1. Zákazník si v prostředí přihlašovacího formuláře vyplní svou registrační e-mailovou adresu. 2. Klikne na odkaz pro obnovu hesla. 3. Aplikace vygeneruje nové heslo k účtu a zašle jej na e-mail zákazníka.
Alternativní scénáře:	Alternativní scénář navazuje na krok 2 hl. scénáře 1. Aplikace identifikuje chybnou e-mailovou adresu. 2. Vyhodí chybovou hlášku a vyzve zákazníka ke znovu-zadání e-mailové adresy.
Výstupní podmínky:	Heslo k zákaznickému účtu bylo obnoveno.
Poznámka:	

Tabulka 4.28: Popis případu užití obnova hesla

Případ užití: Přihlášení zákazníka do systému	
Id:	UC29
Stručný popis:	Zákazník se přihlásí do rezervačního systému.
Aktéři:	Zákazník
Výchozí podmínky:	Uživatel musí mít roli zákazník.
Hlavní scénář:	<ol style="list-style-type: none"> 1. Zákazník vyplní přihlašovací údaje do formuláře na hlavní straně rezervačního systému. 2. Klikne na tlačítko přihlášení. 3. Aplikace přihlásí zákazníka a zobrazí hlavní stránku jeho uživatelského účtu.
Alternativní scénáře:	Navazuje na krok 2 hlavního scénáře <ol style="list-style-type: none"> 1. Přihlašovací údaje jsou zákazníkem vyplněny nesprávně. 2. Aplikace vyhodí chybovou hlášku a vyzve zákazníka k opětovnému zadání údajů.
Výstupní podmínky:	Zákazník byl přihlášen do systému.
Poznámka:	

Tabulka 4.29: Popis případu užití přihlášení zákazníka

Případ užití: Odhlášení zákazníka ze systému	
Id:	UC30
Stručný popis:	Odhlásí zákazníka ze systému.
Aktéři:	Zákazník
Výchozí podmínky:	Uživatel musí mít roli zákazník. Uživatel musí být přihlášený do systému.
Hlavní scénář:	<ol style="list-style-type: none"> 1. Zákazník na stránce svého účtu klikne na možnost odhlásit se. 2. Aplikace odhlásí uživatele ze systému.
Alternativní scénáře:	žádné
Výstupní podmínky:	Uživatel byl odhlášen ze systému.
Poznámka:	

Tabulka 4.30: Popis případu užití odhlášení zákazníka

Případ užití: Vytvoření nové rezervace	
Id:	UC31
Stručný popis:	Zákazník si vytvoří na svém profilu novou rezervaci.
Akteři:	Zákazník
Výchozí podmínky:	Uživatel musí mít roli zákazník. Vybraná rezervovaná jednotka musí být dostupná.
Hlavní scénář:	<ol style="list-style-type: none"> 1. Zákazník si v prostředí rezervačního systému vybere v harmonogramu dané služby jednotku, na kterou se chce přihlásit 2. Klikne na vybranou jednotku. 3. Aplikace označí vybranou jednotku jako rezervovanou a zároveň zobrazí dialogové okno, ve kterém vyzve zákazníka k výběru formy zaplacení rezervované služby. Pokud daná služba obsahuje nějaké podslužby, zobrazí rovněž možnost výběru z těchto podslužeb. 4. Zákazník vybere formu zaplacení (případně zvolí podslužbu) a potvrdí dialogové okno. 4. Klikne na tlačítko potvrzení rezervace. 5. Aplikace ověří dostupnost kreditu na permanentce a vytvoří rezervaci.
Alternativní scénáře:	<p>Alternativní scénář 1 (navazuje na krok 4. hlavního scénáře)</p> <ol style="list-style-type: none"> 1. Aplikace ověří, zda je dostatek kreditu na permanentce. 2. Pokud je kredit nedostatečný, aplikace zamezí vytvoření rezervace a vyhodí chybovou hlášku.
Výstupní podmínky:	Byla vytvořena nová rezervace.
Poznámka:	Pokud je na danou jednotku již vyčerpána přihlašovací kapacita, pole s danou jednotkou bude neaktivní a bude barevně vyznačeno.

Tabulka 4.31: Popis případu užití vytvoření nové rezervace

Případ užití: Storno rezervace	
Id:	UC32
Stručný popis:	Zruší aktivní rezervaci zákazníka.
Akteři:	Zákazník
Výchozí podmínky:	Uživatel musí mít roli zákazník.
Hlavní scénář:	<ol style="list-style-type: none"> 1. Zákazník si na svém účtu zobrazí aktivní rezervace. 2. U vybrané rezervace klikne na tlačítko pro storno. 3. Aplikace stornuje rezervaci, jestliže jsou splněny časové podmínky zrušení rezervace.
Alternativní scénáře:	Žádné
Výstupní podmínky:	Rezervace byla zrušena.
Poznámka:	Časové podmínky pro zrušení rezervace si zadává sportcentrum a jedná se o dobu, kterou je potřeba provést storno rezervace, aby mohla být zrušena (např. je nutné ji zrušit 24 hodin předem). Pokud tato podmínka splněna nebude, storno nebude možno provést (tlačítko pro zrušení bude neaktivní) a zákazník si musí vyjednat případné zrušení telefonicky.

Tabulka 4.32: Popis případu užití registrace zákaznického účtu

Případ užití: Zobrazení rezervací	
Id:	UC33
Stručný popis:	Zobrazí všechny aktivní i neaktivní (vybrané) rezervace.
Aktéři:	Zákazník
Výchozí podmínky:	Uživatel musí mít roli zákazník.
Hlavní scénář:	1. Zákazník si na svém profilu klikne na záložku „moje rezervace”. 2. Aplikace zobrazí aktuální rezervace uživatele.
Alternativní scénáře:	Žádné
Výstupní podmínky:	Zobrazení aktuálních i historických rezervací.
Poznámka:	

Tabulka 4.33: Popis případu užití zobrazení rezervací

Případ užití: Zobrazení rezervací	
Id:	UC33
Stručný popis:	Zobrazí všechny aktivní i neaktivní (vybrané) rezervace.
Aktéři:	Zákazník
Výchozí podmínky:	Uživatel musí mít roli zákazník.
Hlavní scénář:	1. Zákazník si na svém profilu klikne na záložku „moje rezervace”. 2. Aplikace zobrazí aktuální rezervace uživatele.
Alternativní scénáře:	Žádné
Výstupní podmínky:	Zobrazení aktuálních i historických rezervací.
Poznámka:	

Tabulka 4.34: Popis případu užití zobrazení rezervací

d) Matice sledovanosti požadavků

Na následující matici je znázorněno, jak jsou jednotlivé požadavky, uvedené v kapitole 3.3, řešeny v případech užití. Na této matici si lze zkontrolovat, zda je každý požadavek řešen alespoň jedním případem užití.

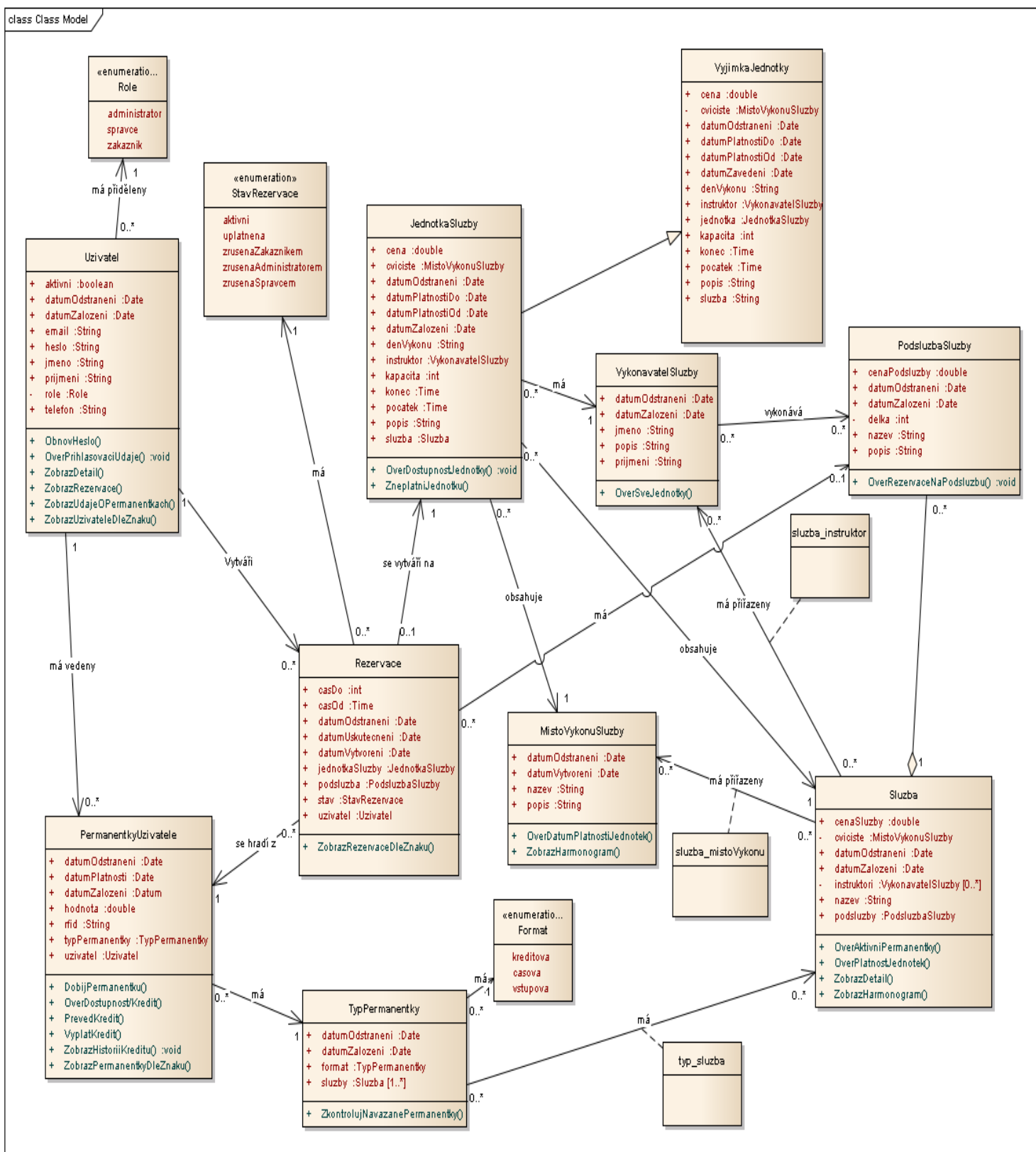
	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9	UC10	UC11	UC12	UC13	UC14	UC15	UC16	UC17	UC18	UC19	UC20	UC21	UC22	UC23	UC24	UC25	UC26	UC27	UC28	UC29	UC30	UC31	UC32	UC33	UC34
P1																									X									
P2															X													X	X	X				
P3																																X	X	
P4																															X			
P5																															X			
P6																																X		
P7																																	X	X
P8																																	X	
P9																										X	X							
P10	X	X				X	X																			X	X							
P11					X																													
P12			X																															
P13					X																													
P14							X																											
P15								X	X	X	X																							
P16																												X	X					
P17												X	X																					
P18												X																						
P19															X																			
P20																								X	X									
P21													X																					
P22														X	X			X	X	X	X													
P23																X																		
P24																					X													
P25																										X	X	X				X	X	X

Obrázek 4.4: Matice trasovatelnosti uživatelských požadavků

4.2 Návrh tříd

S využitím metody analýzy podstatných jmen a sloves byl z případů užití odvoze následující diagram tříd. V diagramu jsou znázorněny třídy i s veškerými potřebnými atributy a jejich datovými typy. Metody, označené v diagramu jsou pouze ty, které jsou specifické pro konkrétní třídy. Každá třída má samozřejmě také metody pro uložení a odstranění, popřípadě zobrazení. Některé třídy také odesílají notificační e-maily, pokud se v system provede akce, která se přímo dotýká rezervací či permanentek zákazníků (viz diagram případů užití). V diagram jsou dále zobrazeny a popsány vztahy mezi jednotlivými třídami s využitím asociace či generalizace nebo agregace, u nichž je zaznamenána multiplicita.

V případě, že se mezi jednotlivými třídami vyskytují vzájemné vazby s multiplicitou m:n, je vazba doplněna o asociační třídu. Třídy s názvem <<enumeration>> vyjadřují, že hodnota daného atributu je tvořena výběrem z možných variant (viz například role, které mohou být v zásadě tři – administrátor, správce pobočky a zákazník).



Obrázek 4.5: Diagram tříd

Obsah

Obecné atributy

Těmito atributy disponuje každá třída. Datum založení se ukládá automaticky při založení nového objektu dané třídy. Datum odstranění se automaticky ukládá při odstranění objektu dané třídy. Odstranění znamená odstranění pouze ve smyslu „smazání z aplikace“. Fyzické odstranění z databáze touto formou neprobíhá. Je-li přítomno datum odstranění, indikuje to odstranění objektu. Pokud datum odstranění není přítomno, objekt je aktivní.

Atribut	Datový typ	povinný	komentář
datumZalozeni/Vytvoreni/Zavedeni	Date	ano	Automatické zadání datumu v době vytváření objektu.
datumOdstraneni	Date	ne	Automatické zadání datumu v době odstranění objektu. Pokud je toto datum zadáno, indikuje odstraněný objekt.

Tabulka 4.35: Obecné atributy

Uživatel

Třída uživatel charakterizuje uživatele systému. Jedná se jak o zákazníky, tak i administrátory a správce poboček. Zda se jedná o zákazníka nebo administrátora či správce se definuje na základě přiřazených rolí. Tyto role se v případě zákazníka přiřadí při klasické registraci automaticky, nebo je v případě registrace přes administrátorské rozhraní přidělí administrátor.

Atribut	Datový typ	povinný	komentář
e-mail	String	ano	E-mail uživatele, který slouží zároveň jako login a jednoznačně identifikuje uživatele systému.
heslo	String	ano	Heslo představuje řetězec znaků, zadaných uživatelem, který slouží k ověření identity při přihlašování uživatelů do systému.
jmeno	String	ano	Jméno uživatele, které se zadává při registraci do systému.
prijmeni	String	ano	Příjmení uživatele, které se zadává do systému při registraci.
telefon	String	ano	Telefonní kontakt na uživatele.
aktivni	boolean	ano	Vyjadřuje, zda je uživatel aktivní či není. Aktivní uživatel může vstupovat do systému, prohlížet údaje na svém profilu a tvořit rezervace na libovolné služby. Oproti tomu pokud je uživatel neaktivní, znamená to, že z nějakého důvodu mu bylo administrátorem

			či správcem zabráněno vytvářet rezervace na služby. Takovýto uživatel si tedy může pouze prohlížet údaje na profilu, ale není mu umožněno tvořit rezervace.
role	enum	ano	Role uživatele, která říká, jaké má daný uživatel v systému pravomoci. Dostupnými rolemi jsou administrátor, správce pobočky a zákazník.

Tabulka 4.36: Atributy třídy uživatel

Služba

Třída služba zastupuje sporty či aktivity, které centrum poskytuje ve své nabídce. Každá tato služba má základní cenu. Cena služby je nepovinným atributem, jelikož služby typu masáže nemohou být jednotně oceněny. Obsahují totiž podslužby (jednotlivé typy masáží). Cena dané služby se pak odvíjí od výběru podslužby.

Atribut	Datový typ	povinný	komentář
cenaSluzby	double	ne	Základní cena dané služby.
nazev	String	ano	Název služby.

Tabulka 4.37: Atributy třídy služba

Místo výkonu služby

Místo výkonu služby je třída, která zastupuje jednotlivé tělocvičny či masérny, které má sportovní centrum k dispozici a ve kterých probíhají veškeré služby.

Atribut	Datový typ	povinný	komentář
nazev	String	ano	Název místa výkonu, například Aerobní sál 1 či Spinningový sál.
popis	String	ne	Do popisu je možné uvést libovolné další informace o konkrétním místě výkonu.

Tabulka 4.38: Atributy třídy místo výkonu služby

Vykonavatel služby

Vykonavatelé služeb jsou trenéři či maséři, kteří jsou zaměstnaní ve sportovním centru a kteří vedou lekce poskytovaných služeb či služby přímo vykonávají (maséři). Každý vykonavatel, resp. instruktor může vést více různých lekcí.

Atribut	Datový typ	povinný	komentář
jmeno	String	ano	Křestní jméno vykonavatele.
prijmeni	String	ano	Příjmení vykonavatele.
popis	String	ne	Další informace o daném vykonavateli.

Tabulka 4.39: Atributy třídy vykonavatel služby

Podslužba služby

Podslužby jsou třídou, která znázorňuje specifický druh služeb, jež si vybírá samotný zákazník přímo při rezervaci. Takovouto službou, která obsahuje podslužby jsou v tomto případě masáže, podslužbami je pak například masáž šíje či celková masáž. Od vybrané podslužby se pak odvíjí délka i cena celé rezervace.

Atribut	Datový typ	povinný	komentář
nazev	String	ano	Název podslužby, například „Masáž šíje“.
cenaPodsluzby	double	ano	Cena podslužby.
delka	int	ano	Délka podslužby, uváděná v minutách, například zádová masáž má délku 30 min.
popis	String	ne	Dodatečné informace o dané podslužbě.

Tabulka 4.40: Atributy třídy podslužba služby

Jednotka služby

Jednotka služby je konkrétní časový úsek, na který se již přímo lze zarezervovat. Jednotkou služby může být například hodina spinningu v pondělí od 17:00 do 18:00, kterou vede instruktor Petr a na kterou se může přihlásit 20 zákazníků.

Atribut	Datový typ	povinný	komentář
cena	double	ano	Cena jednotky. Výchozí cenou jednotky může být cena služby, ovšem každá jednotka může mít upravenou cenu, například pokud by sportovní centrum chtělo o víkendu jiné ceny než přes týden, cenu si u víkendových jednotek při zadávání upraví podle svých preferencí.
datumPlatnostiOd	Date	ano	Datum, od kterého sportovní centrum danou jednotku služby bude provozovat.
datumPlatnostiDo	Date	ano	Datum do kdy je daná jednotka platná.
denVykonu	String	ano	Den v týdnu, kdy bude daná jednotky služby provozována.

kapacita	int	ano	Kapacita dané jednotky služby. Kapacita se například při provozu spinningu odvíjí od počtu dostupných spinningových kol ve spinningovém sále. Dále se může odvíjet od prostorové náročnosti provozovaného cvičení apod.
pocatek	Time	ano	Čas počátku jednotky.
konec	Time	ano	Čas konce jednotky.
popis	String	ne	Dodatečný popis jednotky, například zda je pro začátečníky nebo pokročilé apod.

Tabulka 4.41: Atributy třídy jednotka služby

Výjimka jednotky

Výjimka jednotky je třída, která dědí své atributy z třídy Jednotka služby, proto je v diagramu k jednotce služby připojena vazbou generalizace. Tato třída znázorňuje výjimky, které lze udělit jednotlivým jednotkám. Řeší situace, jako například dočasná změna ceny, dočasná změna instruktora, dočasná změna termínu dané jednotky apod. Jedná se tedy pouze o dočasné úpravy v případě onemocnění instruktora, rozdílných cen o svátcích apod.

Atribut	Datový typ	povinný	komentář
cena	double	ne	Cena jednotky, kterou sportcentrum potřebuje změnit na určitou konkrétní dobu (například v době vánoc chce zdrazit ceny).
datumPlatnostiOd	Date	ne	Datum, od kterého platí výjimka služby.
datumPlatnostiDo	Date	ne	Datum, do kterého platí výjimka služby.
denVykonu	String	ne	Slouží k nutné změně dne výkonu, když chce sportcentrum danou jednotku například dočasně přesunout z pondělí na středu.
kapacita	int	ne	Dočasná změna kapacity jednotky, např. při pokažení spinningového kola je nutné kapacitu o dané kolo snížit, než se opraví.
pocatek	Time	ne	Změna počátečního času jednotky.
konec	Time	ne	Změna času konce jednotky.
popis	String	ne	Dodatečná změna popisu, například pokud se následující měsíc bude místo začátečníků cvičit pokročilé cvičení.

R Tabulka 4.42: Atributy třídy výjimka jednotky

Rezervaci provádí zákazník na konkrétní jednotky vybraných služeb.

Atribut	Datový typ	povinný	komentář
datumUskuteneni	Date	ano	Datum, na které je rezervace plánována.
casOd	Time	ano	Čas počátku rezervace.
casDo	Time	ano	Čas konce rezervace,
stav	enum	ano	Stav rezervace.

Tabulka 4.43: Atributy třídy rezervace

Typy permanentek

Typ permanentky se vždy zadává na určitou službu. Umožňuje vést permanentky na konkrétní služby ve vybraném formátu.

Atribut	Datový typ	povinný	komentář
format	enum	ano	Formát znamená, zda je daná permanentka platná na časový úsek (časová), na počet vstupů (vstupová) či na množství kreditu (kreditová).

Tabulka 4.44: Atributy třídy typy permanentek

Permanentky uživatele

Permanentky uživatele zastupují již každou specifickou permanentku, která je přiřazena ke konkrétnímu zákazníkovi.

Atribut	Datový typ	povinný	komentář
datumPlatnosti	Date	ano	Datum, do kdy daná permanentka uživateli platí.
hodnota	double	ano	Hodnota, na jakou je permanentka nabitá. Může být buď na počet vstupů nebo nabitý kredit. Pokud je permanentka časového typu, výchozí hodnota kreditu je 0kč a není možné tuto permanentku dobít.
rfid	String	ne	Kód čipové karty v případě integrace rezervačního systému s již existujícím systémem čipových karet.

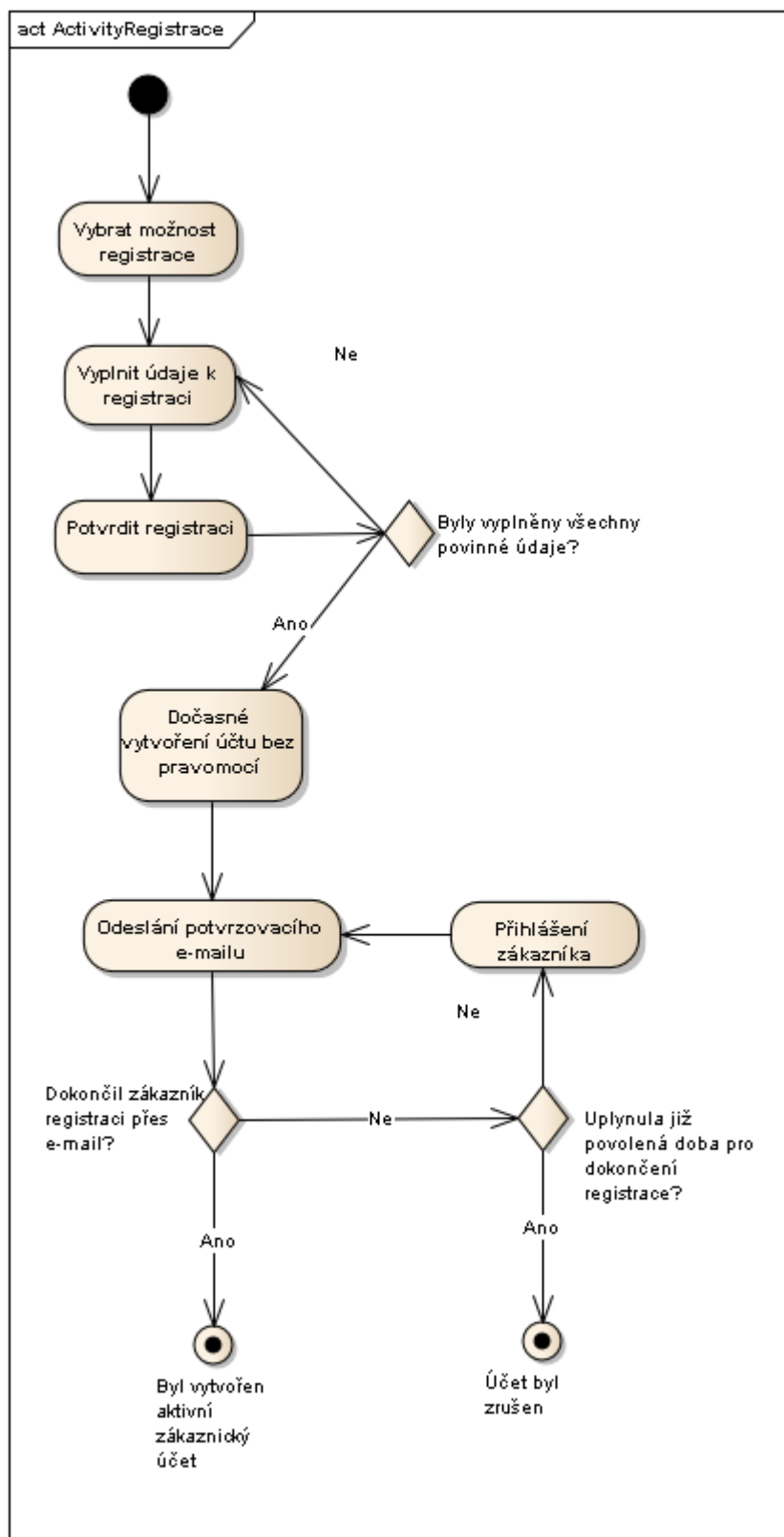
Tabulka 4.45: Atributy třídy permanentky uživatele

4.3 Diagramy aktivit

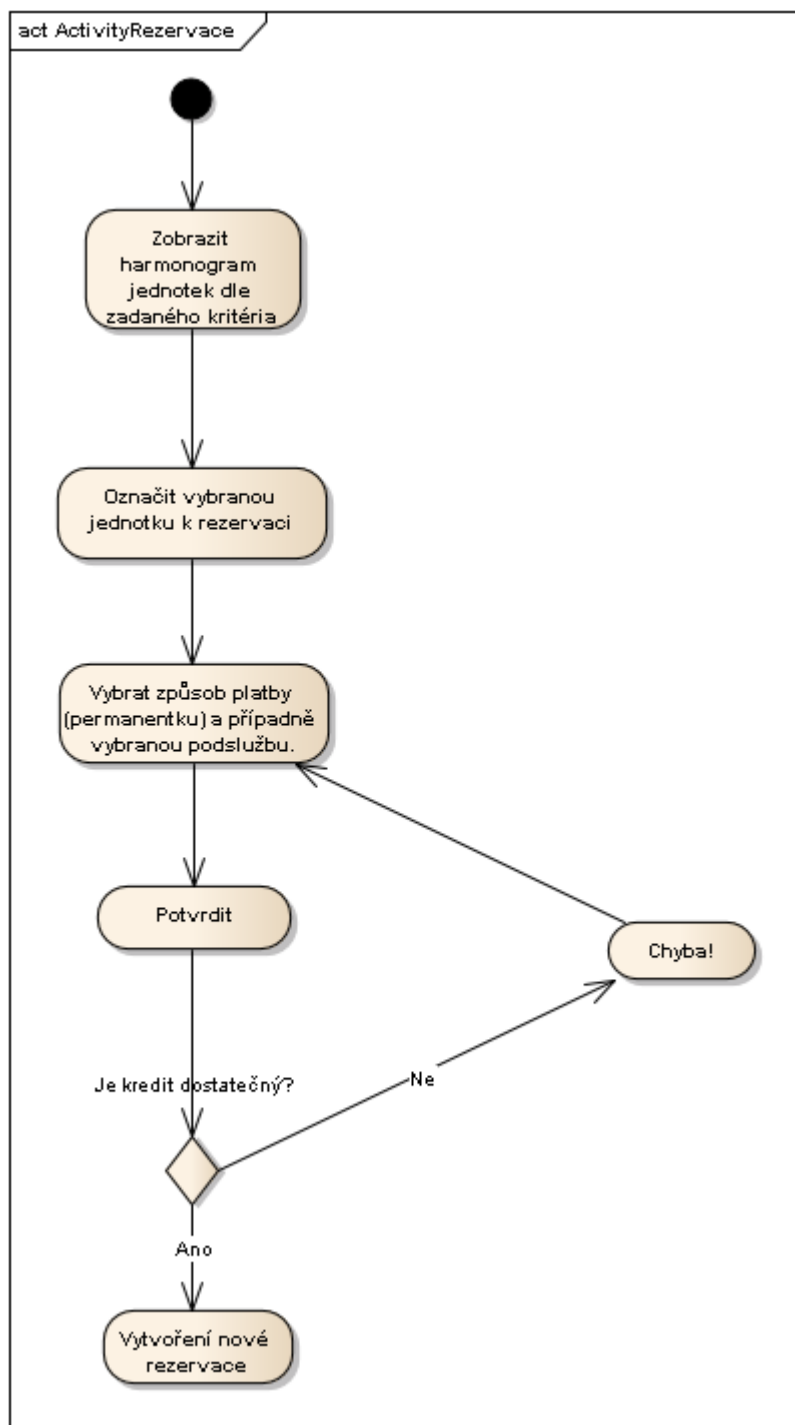
Jak již bylo zmíněno v druhé kapitole, diagramy aktivit napomáhají graficky znázornit scénáře složitějších případů užití. Pro tento účel byly vybrány dva případy užití, které měly kromě hlavního scénáře vždy alespoň jeden alternativní scénář. Těmito případy užití jsou registrace zákazníka do systému a vytvoření nové rezervace zákazníkem.

Při registraci zákazníka do systému musí být splněny některé podstatné podmínky, bez nichž nelze registraci dokončit. Za prvé je nutné vyplnit všechny povinné údaje pro vytvoření nového účtu. Účet zkrátka nemůže existovat bez přihlašovacího jména (resp. e-mailové adresy) a hesla. Dalšími povinnými údaji může být i jméno a příjmení či telefonní číslo. Které z údajů budou povinné a které ne, to závisí čistě na preferenci majitele sportovního centra. Druhou důležitou podmínkou pro úspěšné dokončení registrace je potvrdit odkaz, který bude zaslán zákazníkovi na jím uvedenou e-mailovou adresu, zadanou při registraci. Do doby, než zákazník potvrdí kliknutím na odkaz v e-mailu svou registraci do systému, jeho účet bude neaktivní. Na tento účet se zákazník bude moci přihlásit a bude moci si prohlížet prostředí zákaznického účtu, nebude ovšem schopen provést jakoukoli rezervaci. Na potvrzení registrace bude mít zákazník předem stanovenou dobu, kterou si sportovní centrum samo zvolí. Po uplynutí této doby bude zákaznický účet samovolně odstraněn. Průběh registrace znázorňuje obrázek 4.6: diagram aktivit pro registraci zákaznického účtu.

Druhým případem užití je vytvoření rezervace. Tento případ užití má rovněž podmínku, na základě které se může scénář odvíjet dvěma směry. Pro vytvoření rezervace je nezbytně nutné, aby po výběru permanentky, ze které si zákazník přeje rezervaci uhradit, byl na této permanentce dostatečný kredit. Pokud systém při ověření zjistí nedostatek kreditu, bude zákazník vyzván k opětovnému vybrání jiné permanentky a rezervace nebude vytvořena. Pokud je kredit dostatečný, rezervace bude dokončena bez problému. Tento případ užití znázorňuje obrázek 4.7: Diagram aktivit pro vytvoření rezervace.



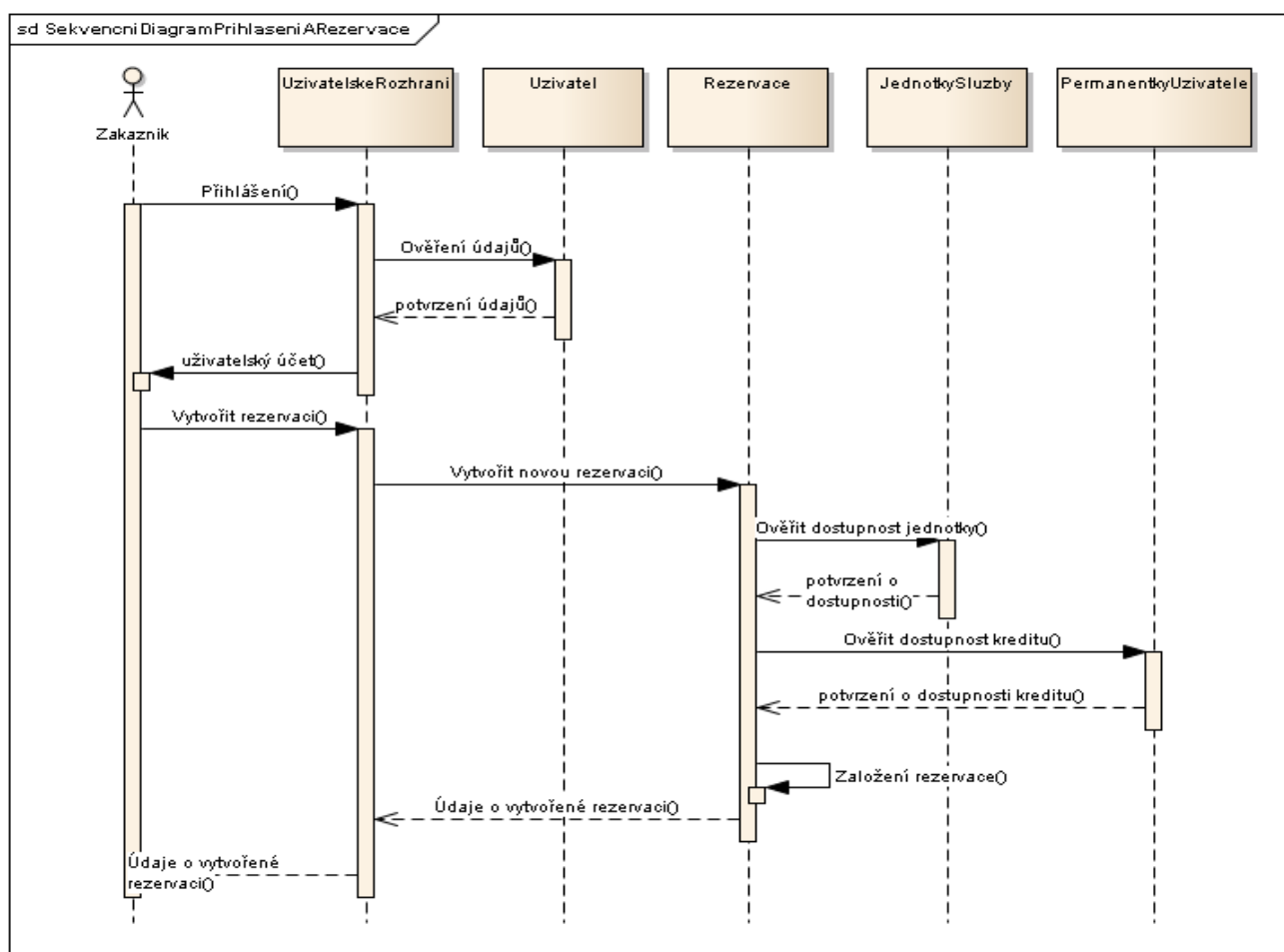
Obrázek 4.6: Diagram aktivit pro registraci zákaznického účtu



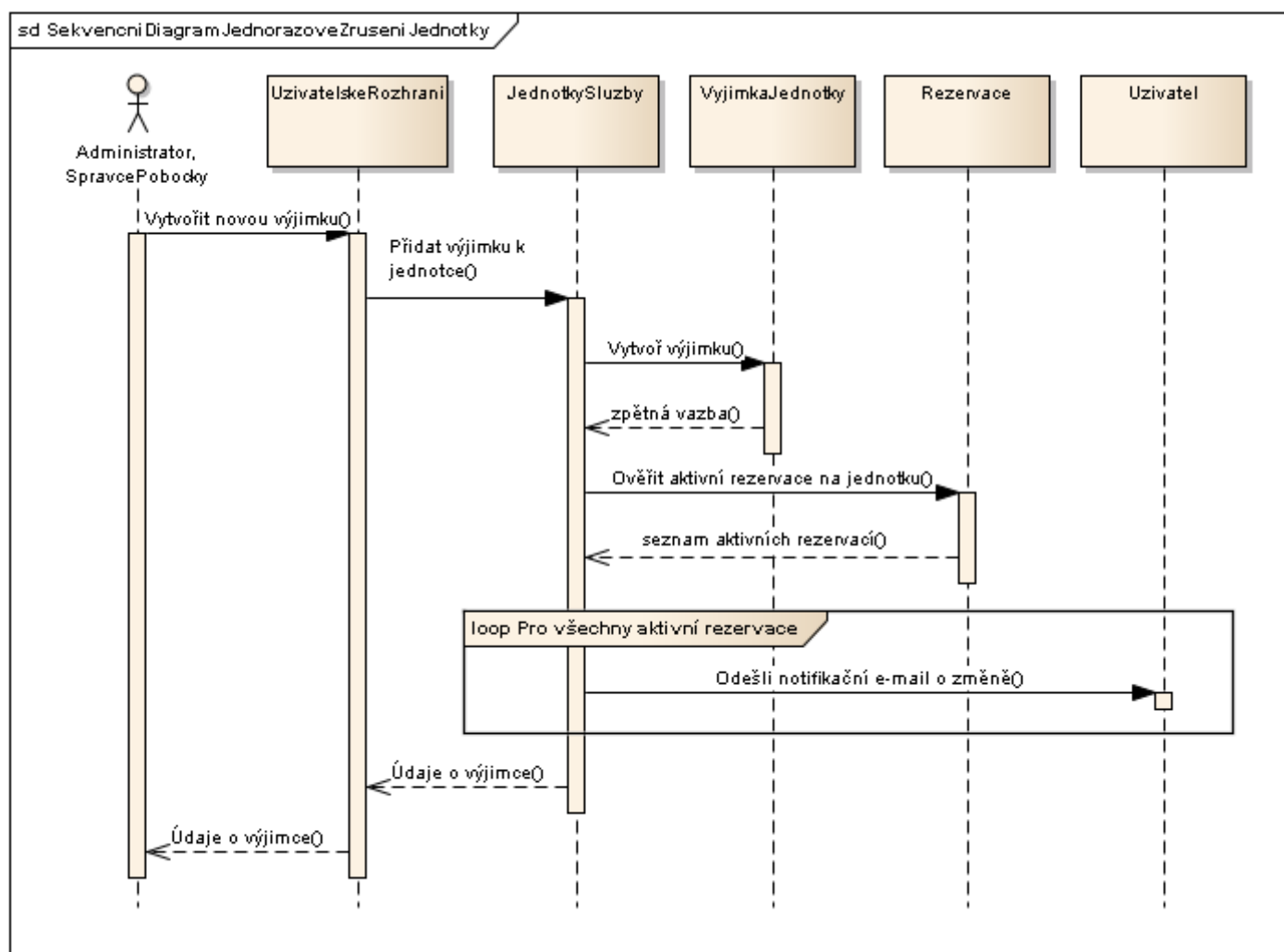
Obrázek 4.7: Diagram aktivit pro vytvoření rezervace

4.4 Sekvenční diagramy

Sekvenční diagramy zobrazují, jak si mezi sebou jednotlivé třídy posílají zprávy, resp. jak spolu komunikují při vykonávání jednoho či několika souvisejících případů užití. Pro tento účel byly vybrány případy užití přihlášení a tvorby rezervace, které na sebe mohou bezprostředně navazovat. Rezervaci bez přihlášení vytvořit nejde, přihlášení je tedy výchozí podmínkou pro její tvorbu. Na obrázku 4.8: Sekvenční diagram přihlášení a rezervace lze pozorovat, jak si třídy mezi sebou posílají zprávy, jak se ověřují údaje, potřebné k přihlášení a tvorbě rezervací a jakou roli v tomto hraje zákazník jako aktér. Jak lze na obrázku vidět, zákazník komunikuje s jednotlivými třídami prostřednictvím uživatelského rozhraní. Druhý diagram vyjadřuje posloupnost interakcí v případě tvorby výjimky jednotky služby. O každé takovéto změně musí být informován každý zákazník, který je na danou jednotku rezervován. V diagramu lze tedy pozorovat i to, jak budou zákazníkům odesílány notifikační e-maily o změně. (viz obrázek 4.9: Sekvenční diagram tvorby výjimky jednotky)



Obrázek 4.8: Sekvenční diagram pro přihlášení a tvorbu rezervace



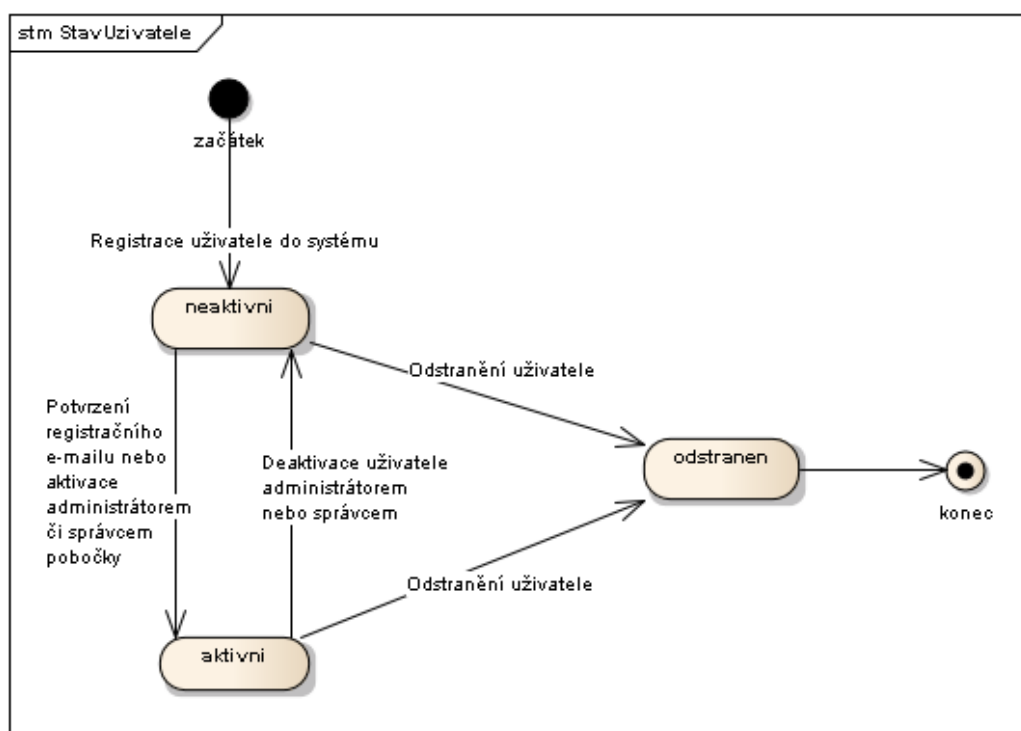
Obrázek 4.9: Sekvenční diagram pro tvorbu výjimky jednotky

4.5 Diagramy stavového automatu

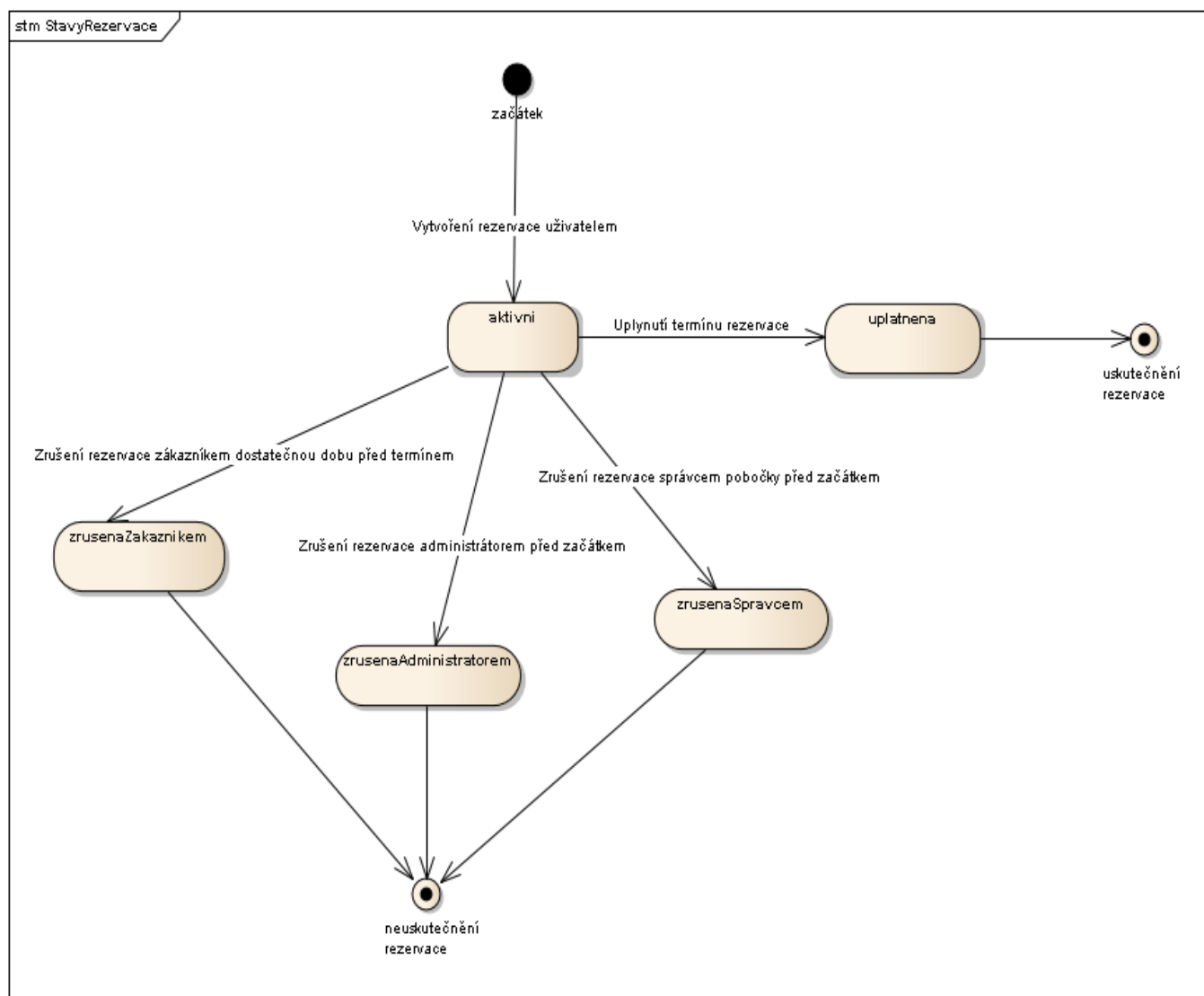
Stavové automaty pomáhají vyjádřit stavy jednotlivých tříd (pokud dané třídy nějakými stavy prochází v průběhu svého životního cyklu) a události, které jednotlivé stavy zapříčiňují. V případě modelu rezervačního systému sportcentra je takovouto třídou třída rezervace, případně uživatel.

Jistými stavy může procházet i uživatel. Ten může být buď neaktivní, a to v době kdy je registrovaný, ale registrace ještě nebyla potvrzena. Po potvrzení se stává aktivním. Tento stav má oprávnění změnit administrátor nebo správce například v situaci, kdy se zákazník nechová v souladu se stanovenými pravidly sportcentra. V případě odstranění uživatelského účtu se pak uživatel nachází ve stavu odstraněn. Tuto posloupnost akcí a změn stavů znázorňuje obrázek 4.10: Stavový automat uživatele.

Třída rezervace může procházet několika stavy. Rezervace je aktivní, pokud je vytvořená a čeká na uplatnění. Je tedy aktivní v době mezi vytvořením a dobou, kdy daná služba proběhne. Po uplynutí této doby se rezervace stává uplatněnou. Rezervaci lze rovněž zrušit. Na možnosti zrušení má vliv uživatel, resp. jeho role. Stav rezervace je po takovémto úkonu zrušená. Přejchod těchto stavů znázorňuje obrázek 4.11: Stavový automat rezervace.



Obrázek 4.10: Stavový automat pro uživatele



Obrázek 4.11: Stavový automat pro rezervaci

4.6 Návrh uživatelského rozhraní rezervačního systému

V průběhu návrhu systému byly vytvářeny návrhy grafického uživatelského rozhraní. Tyto návrhy napomáhaly již ve fázi identifikace jednotlivých případů užití. Díky nim bylo možné postupně upřesňovat případy užití, či dokonce přidávat nové. Tyto návrhy byly rozděleny rovněž do tří skupin, podle dělení případů užití na návrhy rozhraní zákaznických scénářů, administrace uživatelských účtů a základní administraci sportcentra.

Návrhy rozhraní zákaznických scénářů vyobrazují uživatelské rozhraní, přes které se zákazník registruje, přihlašuje či vytváří rezervace a manipuluje se svým účtem. Celkem byly vytvořeny tři návrhy. První z nich zobrazuje, co zákazník, či uživatel obecně uvidí při vstupu na rezervační systém, viz příloha č. 1. Následující návrh vyobrazuje situaci po přihlášení do systému zákazníkem (příloha č. 2). Poslední návrh zákaznických scénářů, viz příloha č. 3 je pohled na záložku “moje rezervace”, kde si zákazník může prohlédnout své aktuální i historické rezervace.

Další skupinou návrhu rozhraní je základní administrace sportcentra. První návrh ukazuje, jak bude vypadat nabídka po přihlášení uživatele s rolí administrátor nebo správce pobočky (příloha č. 4). Po kliknutí na možnost rezervace budou další kroky shodné s návrhy rozhraní zákaznických scénářů. Druhá možnost je možnost administrace. Po kliknutí na nabídku administrace se zobrazí menu s příslušnými položkami, které se administrace týkají, viz příloha č. 5. Další návrh zobrazuje administrativní správu typů permanentek a vyhledávání již konkrétních permanentek zákazníků (příloha č. 6). Poslední dva návrhy ukazují administraci služeb a pak náhled konkrétní služby (přílohy č. 7 a 8).

Poslední skupinou je skupina návrhů rozhraní administrace uživatelských účtů. V tomto prostředí se spravují veškeré uživatelské účty a vše, co s nimi souvisí. Tato skupina obsahuje opět celkem tři návrhy. První je vzorový náhled na správu uživatelských účtů, kde lze přidat nový účet či vyhledat existující účty podle zadaných kritérií (příloha č. 9). Další dva návrhy se zaměřují na detail uživatelského účtu v záložce permanentky a jejich editaci (příloha č. 10 a 11).

5. Závěr

Cílem této diplomové práce bylo analyzovat výchozí situaci a navrhnout rezervační systém pro podnikatele v oblasti sportu a relaxace. Ihned v úvodu byl proveden sběr uživatelských požadavků formou rozhovoru. Tato metoda byla doplněna o analýzu dokumentů, jejímž hlavním zdrojem byly webové stránky sportovního centra, kde jsou dostatečně popsány všechny služby, které centrum nabízí. Po získání dostatečného množství informací byla provedena analýza potřeb sportovního centra a jeho služeb. Na tuto analýzu plynule navázala tvorba případů užití za pomoci jazyka UML 2.0, kdy byly identifikovány funkce systému, vycházející z požadavků sportovního centra. Následně byla provedena lexikální analýza případů užití, která vyústila v návrh diagramu tříd, jímž se modeluje statická struktura systému. Pro lepší pochopení byl návrh systému doplněn o několik diagramů jazyka UML 2.0, kterými jsou diagramy aktivit, sekvenční diagramy a diagramy stavových automatů.

Již v průběhu analýzy uživatelských požadavků byly vytvářeny nefunkční prototypy, které měly za úkol vytvořit představu o vizuální podobě systému. Základem pro tvorbu těchto prototypů byly prvotní vytvořené případy užití. Tyto prototypy pak zároveň napomáhaly přesnější specifikaci případů užití a celé funkcionality systému.

V matici trasovatelnosti požadavků pak bylo ověřeno, že každý zadaný uživatelský požadavek byl součástí alespoň jednoho případu užití, což vypovídá o tom, že v průběhu tvorby systému žádné z těchto požadavků nebyly opomenuty.

Zadaný cíl práce byl splněn a návrh systému je nyní připraven k implementační fázi.

Seznam použité literatury

Knihy a učební texty

ARLOW, Jim; NEUSTADT, Ila. UML 2 a unifikovaný proces vývoje aplikací. Překlad Bogdan Kizska. 1.vyd. Brno: Computer Press, 2007. 567 s. ISBN 978-80-251- 1503-9.

GEORGE, Joey F., Dinesh BATRA, Joseph S. VALACICH a Jeffrey A. HOFFER. Object-oriented Systems Analysis and Design. 2nd ed. New Jersey: Pearson, 2007. 464 s. ISBN 978-0132279000.

GRÄSSLE, Patrick, Henriette BAUMANN a Philippe BAUMANN. UML 2.0 in action - a project based tutorial. Birmingham: Packt Publishing, 2005. 229 s. ISBN 978-1-904811-55-8.

PROCHÁZKA, Jaroslav; KLIMEK, Cyril. Softwarové inženýrství. 2. vyd. Ostrava: Ostravská univerzita v Ostravě, 2009. 132 s.

TVRDÍKOVÁ, Milena. Aplikace moderních informačních technologií v řízení firmy. 1. vyd. Praha: Grada, 2008, s. 1-50. ISBN 978-80-247-2728-8.

VONDRÁK, Ivo. Úvod do softwarového inženýrství. 1. vyd. Ostrava: VŠB- Technická univerzita ostrava, 2002. 74 s.

Internetové zdroje

HAWTIN, Steve. Unified Modelling language. [online]. 2010 [cit. 2013-02-20]. Dostupné z: http://tsort.info/iabok/unified_modeling_language.htm

Objekty – Objektová analýza, návrh a programování. [online]. [cit. 2012-11-22]. Dostupné z: objekty.vse.cz

Object-Oriented Software Engineering – OOSE. [online]. [cit. 2012-11-21]. Dostupné z: <http://cs-exhibitions.uni-klu.ac.at/index.php?id=448>

Object modelling technique – OMT. [online]. [cit. 2012-11-25]. Dostupné z: <http://www.idi.ntnu.no/grupper/su/publ/html/totland/ch0527.htm>

PASTORČÁK, Petr. Objektově orientované modelování systému. [online]. 2004 [cit. 2013-04-10]. Dostupné z: <http://orca.xf.cz/ooms/>

REJNKOVÁ, Petra. Příklady použití diagramů jazyka UML 2.0. [online]. 2009 [cit. 2013-04-12]. Dostupné z: <http://uml.czweb.org/index.html>

RYDVAL, Slávek. OCUP.cz aneb jak si certifikovat znalosti UML 2. [online]. 2011 [cit. 2013-03-05]. Dostupné z: <http://ocup.ocup.cz/2010/11/sekvencni-diagramy.html>

Vývoj metodik a metod objektové analýzy a designu. [online]. 1999 [cit. 2012-11-21]. Dostupné z: http://www.komix.cz/Tisk/Clanky/Historie/Vyvoj_metodik.aspx

Prohlášení o využití výsledků diplomové práce

Prohlašuji, že

- jsem byl(a) seznámen(a) s tím, že na mou diplomovou (bakalářskou) práci se plně vztahuje zákon č. 121/2000 Sb. – autorský zákon, zejména § 35 – užití díla v rámci občanských a náboženských obřadů, v rámci školních představení a užití díla školního a § 60 – školní dílo;
- beru na vědomí, že Vysoká škola báňská – Technická univerzita Ostrava (dále jen VŠB-TUO) má právo nevýdělečně, ke své vnitřní potřebě, diplomovou (bakalářskou) práci užít (§ 35 odst. 3);
- souhlasím s tím, že diplomová (bakalářská) práce bude v elektronické podobě archivována v Ústřední knihovně VŠB-TUO a jeden výtisk bude uložen u vedoucího diplomové (bakalářské) práce. Souhlasím s tím, že bibliografické údaje o diplomové (bakalářské) práci budou zveřejněny v informačním systému VŠB-TUO;
- bylo sjednáno, že s VŠB-TUO, v případě zájmu z její strany, uzavřu licenční smlouvu s oprávněním užít dílo v rozsahu § 12 odst. 4 autorského zákona;
- bylo sjednáno, že užít své dílo, diplomovou (bakalářskou) práci, nebo poskytnout licenci k jejímu využití mohu jen se souhlasem VŠB-TUO, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly VŠB-TUO na vytvoření díla vynaloženy (až do jejich skutečné výše).

V Ostravě dne 26.04.2013



.....

Bc. Pavlína Polanská

Seznam příloh

Příloha č. 1: Zákaznické scénáře - přihlašování

Příloha č. 2: Zákaznické scénáře – přihlášení zákazníka

Příloha č. 3: Zákaznické scénáře – zobrazení rezervací

Příloha č. 4: Základní administrace - přihlášení administrátora/správce

Příloha č. 5: Základní administrace – nabídka administrace

Příloha č. 6: Základní administrace – administrace permanentek

Příloha č. 7: Základní administrace – administrace služeb

Příloha č. 8: Základní administrace – detailní administrace služeb

Příloha č. 9: Administrace uživatelských účtů – administrace uživatelů

Příloha č. 10: Administrace uživatelských účtů – detail uživatele

Příloha č. 11: Administrace uživatelských účtů – správa/editace permanentek uživatele

